Proven COMPUTER Performance

SYSTEMS

A Subsidiary of GOULD INC.

FIRMWARE MANUAL

SYSTEMS 32/75A IPU/CPU

Model 2005

June 1980

305-322005-001

This manual is supplied without representation or warranty of any kind. SYSTEMS therefore assumes no responsibility and shall have no liability of any kind arising from the supply or use of this publication or any material contained herein.

REVISION INSTRUCTIONS AND MANUAL HISTORY

EQUIPMENT: IPU/CPU **PUBLICATION NO.** 305-322005-000

PURPOSE: This printing provides the firmware documentation 531-322675-002 for the Model 2005 IPU/CPU.

REVISION INSTRUCTIONS: Delete and add pages as shown on the following table. File this page in front of page

DELETE	ADD
N/A	N/A

MANUAL HISTORY

*REV. TYPE	REV. NO.	DATE ISSUED	*REV. TYPE	REV. NO.	DATE ISSUED
1st Ptg		1-80			

*I = INTERIM REVISION
F = FORMAL REVISION
R = REISSUE
A = ADDENDUM

APPLICABLE CONTROL DOCUMENTS

The information in this publication pertains to the following equipment at the Equipment Revision Level (ERL) indicated by the last blacked out number in the table.

IPU/CPU

				04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

00	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19
20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59
60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79
80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99

ERN 301-39 DATE 11-27-79 REV Ø

APPLICATION	
NEXT ASSY	USED ON
160-103436	32
160-103438	32

REVISIONS			
LTR	DESCRIPTION	DATE	APPROVED
Ø	Initial Level		
A	80-169 HC 5/14/80 CR	6-11-80	JDS

PART REV 3A 002

REVISION STATUS	SHEET	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26
	REVISION	X	002	001	001	007	006	006	006	006	006	006	006	001	002												

UNLESS OTHERWISE SPECIFIED DIMENSIONS ARE IN INCHES/MM

INCHES

DECIMAL PLACES .XX ± .XXX ±

UNIT PLACES .XX ± .XXX ±

ANGLES ± 0° 30'

DRAWN	Ron Odille	DATE 11-27-79
CHECKER	Ron Odille	11-27-79
ENGINEER	D. W. Kramer	3/14/80
PROJ APPROVED	John R. Flanagan	3-14-80
APPROVED	Steve Hoover	

SYSTEMS Fort Lauderdale, Florida 33313
ENGINEERING LABORATORIES

PROM CONTROL LISTING - INTERNAL PROCESSING UNIT/ CENTRAL PROCESSING UNIT

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675

SCALE None SHEET 1 OF 14

BRUNING 40-22 43780

SYSTEMS Form No. 059E

REVISION CONTROL
IPU/CPU

531-322675	001	002											
531-322676	001	002											
531-322703	001	001											
531-322701	007	007											
531-322504	006	006											
531-322510	006	006											
531-322511	006	006											
531-322512	006	006											
531-322513	006	006											
531-322514	006	006											
531-322515	006	006											
531-322516	006	006											
531-322708	002	002											

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE	REV A	SHEET 2

FIRMWARE ELEMENT: IPU/CPU CONTROL ROM (CROM)

FIRMWARE PRODUCT NO.: 531-322676-001

MICRO-WORD FORMAT (MSB … LSB)

PROM HEX	DESCRIPTION	0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31	32-35	36-39	40-43	44-47
000-3FF	IDENT.	1	2	3	4	5	6	7	8	9	10	11	12
	REV.	001	001	001	001	001	001	001	001	001	001	001	001
400-7FF	IDENT.	13	14	15	16	17	18	19	20	21	22	23	24
	REV.	001	001	001	001	001	001	001	001	001	001	001	001
800-BFF	IDENT.	25	26	27	28	29	30	31	32	33	34	35	36
	REV.	001	001	001	001	001	001	001	001	001	001	001	001
C00-FFF	IDENT.	37	38	39	40	41	42	43	44	45	46	47	48
	REV.	001	001	001	001	001	001	001	001	001	001	001	001

PROM IDENTIFICATION/COORDINATOR LOCATION

160-103438 IPU/CPU

Board "C"

1	2	3	4	5	6	7	8	9	10	11	12
V24	V23	V22	V21	V20	V19	V18	V17	V16	V15	V14	V13
13	14	15	16	17	18	19	20	21	22	23	24
T24	T23	T22	T21	T20	T19	T18	T17	T16	T15	T14	T13
25	26	27	28	29	30	31	32	33	34	35	36
S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13
37	38	39	40	41	42	43	44	45	46	47	48
R24	R23	R22	R21	R20	R19	R18	R17	R16	R15	R14	R13

SIZE A | CODE IDENT NO. 20886 | DWG NO. 531-322675 | REV g | SCALE — | SHEET 3

Form 060C BRUNING - MIAMI 40-81

FIRMWARE ELEMENT: IPU/CPU CONTROL ROM(CROM)

FIRMWARE PRODUCT NO.: 531-322676-002

MICRO-WORD FORMAT

PROM HEX	DESCRIPTION	MSB 0-3	4-7	8-11	12-15	16-19	20-23	24-27	28-31	32-35	36-39	40-43	LSB 44-47
000-3FF	IDENT.	1	2	3	4	5	6	7	8	9	10	11	12
	REV.	002	002	002	002	002	002	002	002	002	002	002	002
400-7FF	IDENT.	13	14	15	16	17	18	19	20	21	22	23	24
	REV.	001	001	001	002	001	001	001	001	001	001	001	001
800-BFF	IDENT.	25	26	27	28	29	30	31	32	33	34	35	36
	REV.	001	001	001	001	001	001	001	001	001	001	001	002
C00-FFF	IDENT.	37	38	39	40	41	42	43	44	45	46	47	48
	REV.	002	002	002	002	002	002	002	002	002	002	002	002

SIZE A | CODE IDENT NO. 20886 | DWG NO.

SCALE —/— | REV A | SHEET 3a

160-103438 IPU/CPU BOARD "C"

PROM IDENTIFICATION/COORDINATOR LOCATION

1	2	3	4	5	6	7	8	9	10	11	12
V24	V23	V22	V21	V20	V19	V18	V17	V16	V15	V14	V13
13	14	15	16	17	18	19	20	21	22	23	24
T24	T23	T22	T21	T20	T19	T18	T17	T16	T15	T14	T13
25	26	27	28	29	30	31	32	33	34	35	36
S24	S23	S22	S21	S20	S19	S18	S17	S16	S15	S14	S13
37	38	39	40	41	42	43	44	45	46	47	48
R24	R23	R22	R21	R20	R19	R18	R17	R16	R15	R14	R13

FIRMWARE ELEMENT IPU/CPU PAGE SELECT DECODE (DECODE PROM)

FIRMWARE PRODUCT NO. 531-322703-001

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	001	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103438 IPU/CPU Board "C"

1	2
F09	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 4

Form 060C

FIRMWARE ELEMENT: IPU/CPU DECODE ROMS

FIRMWARE PRODUCT NO.: 531-322701-007

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB							LSB
000-1FF	IDENT.	1	2	3	4	5	6	7	8
	REV.	007	007	007	007	007	007		
	IDENT.	9	10	11	12	13	14	15	16
	REV.								

PROM IDENTIFICATION / COORDINATE LOCATION

160-103438 IPU/CPU Board "C"

1	2	3	4	5	6	7	8
E07	F07	B08	B09	E06	C09		
9	10	11	12	13	14	15	16

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 5

FIRMWARE ELEMENT IPU/CPU ALU DECODE ROM

FIRMWARE PRODUCT NO. 531-322504-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103438 IPU/CPU

Board "C"

1	2
M01	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 6

Form 060C

FIRMWARE ELEMENT IPU/CPU MULTIPLY ASSIST PROM

FIRMWARE PRODUCT NO. 531-322510-006

MICRO-WORD FORMAT

MSB LSB

PROM HEX ADDRESS	DESCRIPTION	0-3	
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
C04	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 7

Form 060C

FIRMWARE ELEMENT IPU/CPU FLOATING POINT ASSIST PROM

FIRMWARE PRODUCT NO. 531-322511-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
G12	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 8

Form 060C

FIRMWARE ELEMENT IPU/CPU FLOATING POINT ASSIST PROM

FIRMWARE PRODUCT NO. 531-322512-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
K04	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 9

Form 060C

FIRMWARE ELEMENT IPU/CPU SCALE PROM

FIRMWARE PRODUCT NO. 531-322513-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
D05	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 10

Form 060C

FIRMWARE ELEMENT IPU/CPU SCALE PROM

FIRMWARE PRODUCT NO. 531-322514-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
E05	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO
A	20886	531-322675
SCALE None	REV Ø	SHEET 11

Form 060C

FIRMWARE ELEMENT IPU/CPU SCALE PROM

FIRMWARE PRODUCT NO. 531-322515-006

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

	1	2
	E06	

	1	2

	1	2

	1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 12

Form 060C

FIRMWARE ELEMENT IPU/CPU SCALE PROM

FIRMWARE PRODUCT NO. 531-322516-001

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 0-3	LSB
000-1FF	IDENT.	1	
	REV.	006	
	IDENT.		
	REV.		
	IDENT.		
	REV.		
	IDENT.		
	REV.		

PROM IDENTIFICATION / COORDINATE LOCATION

160-103436 IPU/CPU

Board "A"

1	2
F06	

1	2

1	2

1	2

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 13

Form 060C

FIRMWARE ELEMENT: FPU CONTROL ROM (CROM)

FIRMWARE PRODUCT NO.: 531-322708-002

MICRO-WORD FORMAT

PROM HEX ADDRESS	DESCRIPTION	MSB 48-51	52-55	56-59	60-63				LSB
000-3FF	IDENT.	1	2	3	4	5	6	7	8
	REV.	002	002	002	002				
	IDENT.	9	10	11	12	13	14	15	16
	REV.								

PROM IDENTIFICATION / COORDINATE LOCATION

160-103190 FPU
Board "E"
Wirewrap

1	2	3	4	5	6	7	8
R01	R02	R03	R04				
9	10	11	12	13	14	15	16

160-103309 FPU
Board "E"
Multiwire

1	2	3	4	5	6	7	8
R01	R02	R03	R04				
9	10	11	12	13	14	15	16	

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

1	2	3	4	5	6	7	8
9	10	11	12	13	14	15	16

SIZE	CODE IDENT NO.	DWG NO.
A	20886	531-322675
SCALE None	REV Ø	SHEET 14

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFI 32/75A CPU/IPU C75R3
PC MICROWORD

```
                                                                      0001.000
                                                                      0002.000
*********************************************************************0003.000
***     MASTER FIRMWARE CONTROL/REVISION # : 531-322675/002     ***;  0004.000
*********************************************************************0005.000

                                                                      0006.000
*********************************************************************0007.000
***     PART NUMBER/REV : 531-322676/002                        ***   0008.000
***     LOCATIONS       : C BOARD, AS LISTED IN THE CHART BELOW ***   0009.000
***     NAME            : CONTROL ROM (CROM)                    ***   0010.000
***     DATE            : APRIL 30,1980                         ***   0011.000
***     SOURCE FILE     : WHSTPUR4                              ***   0012.000
***     OBJECT FILE     : IPUR4                                 ***   0013.000
*********************************************************************0014.000

                                                                      0015.000
*********************************************************************0016.000
***                        C BOARD                                ***   0017.000
***       CONTROL ROM (CROM) LOCATION/REVISION CHART               ***   0018.000
*********************************************************************0019.000
***             *              COLUMN                 *             ***   0020.000
*********************************************************************0021.000
** SIMPLE ROM *    R    *    S    *    T    *    V    *             ***   0022.000
*********************************************************************0023.000
***    CROM     * 600/  * 800/  * 400/  * 000/  *       BIT         ***   0024.000
*** ADDRESSES   *  FFF  *  FFF  *  7FF  *  3FF  *   POSITIONS       ***   0025.000
*********************************************************************0026.000
***  11  * 13  * /02   * /02   * /01   * /02   *    (44/47)         ***   0027.000
*********************************************************************0028.000
***  10  * 14  * /02   * /01   * /01   * /02   *    (40/43)         ***   0029.000
*********************************************************************0030.000
***  09  * 15  * /02   * /01   * /01   * /02   *    (36/39)         ***   0031.000
*********************************************************************0032.000
***  08  * 16  * /02   * /01   * /01   * /02   *    (32/35          ***   0033.000
*********************************************************************0034.000
***  07  * 17  * /02   * /01   * /01   * /02   *    (28/31)         ***   0035.000
*********************************************************************0036.000
***  06  * 18  * /02   * /01   * /01   * /02   *    (24/27)         ***   0037.000
*********************************************************************0038.000
***  05  * 19  * /02   * /01   * /01   * /02   *    (20/23)         ***   0039.000
*********************************************************************0040.000
***  04  * 20  * /02   * /01   * /01   * /02   *    (16/19)         ***   0041.000
*********************************************************************0042.000
***  03  * 21  * /02   * /01   * /02   * /02   *    (12/15)         ***   0043.000
*********************************************************************0044.000
***  02  * 22  * /02   * /01   * /01   * /02   *    (08/11)         ***   0045.000
*********************************************************************0046.000
***  01  * 23  * /02   * /01   * /01   * /02   *    (04/07)         ***   0047.000
*********************************************************************0048.000
***  00  * 24  * /02   * /01   * /01   * /02   *    (00/03)         ***   0049.000
*********************************************************************0050.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC MICROWORD

```
                                                                                0051.000

                                                                                0052.000
*********************************************************************           0053.000
*** NAME                  : HARDWARE FLOATING POINT UNIT CROM       ***          0054.000
*** PART NUMBER/REV       : 531-322708/002                          ***          0055.000
*** LOCATIONS             : FPU 'E' BOARD AS LISTED IN THE CHART    ***          0056.000
*** DATE                  : FEBUARY 05,1979                         ***          0057.000
*** SOURCE FILE           : WH$IPUR4                                ***          0058.000
*** OBJECT FILE           : IPUR4                                   ***          0059.000
*********************************************************************           0060.000


                                                                                0061.000
*********************************************************************           0062.000
*** FPU 'E' BOARD CONTROL ROM (CROM) LOCATION/REVISION CHART       ***          0063.000
*********************************************************************           0064.000


                                                                                0065.000
************************************************                                0066.000
*** CROM ADDRESS = C00/FFF                     ***                             0067.000
************************************************                                0068.000
***TABLE*COLUMN** ROM *  REV  * BIT POSITIONS ***                               0069.000
************************************************                                0070.000
***  12  *   R  *  1  *  /02  *    (48-51)    ***                               0071.000
************************************************                                0072.000
***  13  *   R  *  2  *  /02  *    (52/55)    ***                               0073.000
************************************************                                0074.000
***  14  *   R  *  3  *  /02  *    (56/59)    ***                               0075.000
************************************************                                0076.000
***  15  *   R  *  4  *  /02  *    (60/63)    ***                               0077.000
************************************************                                0078.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC MICROWORD

```
                                                                        0079.000
***                                                                     0080.000
***             PROGRAM STATUS DOUBLE WORD (PSD) DEFINITION             0081.000
***                                                                     0082.000
***    WORKING PROGRAM STATUS DOUBLE WORD - WORD 1                      0083.000
***                                                                     0084.000
***    0       7 8       15 16       23 24       31                    0085.000
***   ********************************************                      0086.000
***   *         *         *           *          *                      0087.000
***   ********************************************                      0088.000
***    WHERE :                                                          0089.000
***    BIT 0   =PRIVILEGE BIT                                           0090.000
***        1-4 =CC 1-4                                                  0091.000
***          5 =EXTENDED INDEXING MODE                                  0092.000
***          6 =LAST INSTRUCTION EXECUTED WAS IN A RIGHT HALFWORD       0093.000
***          7 =ENABLE ARITHMETIC EXCEPTION (75 MODE ONLY)              0094.000
***       8-12 =NOT USED (MUST BE ZERO)                                 0095.000
***      13-29 =WORD ADDRESS OF THE LOGICAL PROGRAM COUNTER VALUE       0096.000
***         30 =NEXT INSTRUCTION IS IN A RIGHT HALFWORD                 0097.000
***         31 =NOT USED (MAY BE A ZERO OR ONE)                         0098.000
***                                                                     0099.000
***                                                                     0100.000
***    DISPLAY PROGRAM STATUS DOUBLE WORD - WORD 1                      0101.000
***                                                                     0102.000
***    0       7 8       15 16       23 24       31                    0103.000
***   ********************************************                      0104.000
***   *         *         *           *          *                      0105.000
***   ********************************************                      0106.000
***    WHERE :                                                          0107.000
***    BIT 0   =PRIVILEGE BIT                                           0108.000
***        1-4 =CC 1-4                                                  0109.000
***          5 =EXTENDED INDEXING MODE                                  0110.000
***          6 =LAST INSTRUCTION EXECUTED WAS IN A RIGHT HALFWORD       0111.000
***          7 =ENABLE ARITHMETIC EXCEPTION (75 MODE ONLY)              0112.000
***          8 =CPU IS IN THE 75 MODE (DISPLAY ONLY)                    0113.000
***          9 =CPU IS IN THE MAPPED MODE (DISPLAY ONLY)                0114.000
***      10-12 =NOT USED (MUST BE ZERO)                                 0115.000
***      13-29 =WORD ADDRESS OF THE LOGICAL PROGRAM COUNTER VALUE       0116.000
***         30 =NEXT INSTRUCTION IS IN A RIGHT HALFWORD                 0117.000
***         31 =CPU IS IN THE BLOCKED INTERRUPT MODE (DISPLAY ONLY)     0118.000
***                                                                     0119.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC MTCROWORD

```
                                                                        0120.000
***                                                                     0121.000
***    WORKING AND DISPLAY PROGRAM STATUS DOUBLE WORD - WORD 2           0122.000
***                                                                     0123.000
***    0       7 8       15 16       23 24       31                    0124.000
***    ***********************************************                  0125.000
***    *         *         *         *         *                        0126.000
***    ***********************************************                  0127.000
***    WHERE :                                                          0128.000
***    BIT 0-1 =MAP GRANULARITY AS FOLLOWS:                             0129.000
***            0 0 OPERATE UNMAPPED                                     0130.000
***            0 1 OPERATE WITH 8K GRANULARITY                          0131.000
***            1 0 OPERATE WITH 8K GRANULARITY                          0132.000
***            1 1 OPERATE WITH 8K GRANULARITY                          0133.000
***      2-15 =BORROW PROCESS INDEX (BPIX)                              0134.000
***      14   =NOT USED                                                 0135.000
***      15   =RETAIN CURRENT MAP CONTENTS                             0136.000
***      16-17 =EXTERNAL INTERRUPT FLAGS AS FOLLOWS:                    0137.000
***            0 0 OPERATE WITH INTERRUPT LEVEL ACTIVE                  0138.000
***            0 1 OPERATE WITH INTERRUPTS BLOCKED                      0139.000
***            1 0 OPERATE WITH CURRENT BLOCKING MODE                   0140.000
***            1 1 OPERATE WITH CURRENT BLOCKING MODE                   0141.000
***      18-29 =CONTROL PROCESS INDEX (CPIX)                            0142.000
***      30-31 =NOT USED                                                0143.000
***                                                                     0144.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC MICROWORD

```
                                                                                                    0145.000
                        ************************************************************************    0146.000
                        **                    REGISTER EQUATES                                  **    0147.000
                        ************************************************************************    0148.000
                        **                                                                      **    0149.000
                        **                    LOWER BANK                                        **    0150.000
                        **                                                                      **    0151.000
                        ************************************************************************    0152.000
000000000000000A        PCMASK      $EQ     A       CONTENTS = 0007FFFC                             0153.000
                        **                                                                          0154.000
0000000000000009        SIMASK      $EQ     9       CONTENTS = FE000000 (55 & 75 MODE)              0155.000
                        **                                                                          0156.000
000000000000000A        MSCD        $EQ     10      MAP SEGMENT CONTROL DESCRIPTOR                  0157.000
                        **                                                                          0158.000
000000000000000B        TEMP1       $EQ     11                                                      0159.000
000000000000000B        MAP.ADDR    $EQ     11      CURRENT MAP REGISTER ADDRESS                    0160.000
                        *                             IN BITS 12/18                                 0161.000
                        **                                                                          0162.000
000000000000000C        ALEVEL      $EQ     12      HIGHEST ACTIVE INTERRUPT LEVEL                  0163.000
                        **                                                                          0164.000
000000000000000D        TEMP3       $EQ     13      USED BY LF, STF, & DVFD                         0165.000
000000000000000D        MID         $EQ     13      MAP IMAGE DESCRIPTOR                            0166.000
                        **                                                                          0167.000
000000000000000E        TEMP4       $EQ     14      USED BY LF, STF, & DVFD                         0168.000
000000000000000E        BUMP        $EQ     14      INCREMENT VALUE = @00008000                     0169.000
000000000000000E        IPL.FLAGS   $EQ     14      CONTENTS :                                      0170.000
                        **                            BITS 0-3=F  IOCD RECEIVED                     0171.000
                        **                            BIT 16=1  IPL SCRATCH PAD CLEARED             0172.000
                        **                                                                          0173.000
000000000000000F        SCR.ADDR    $EQ     15      SCRATCH PAD ADDR IN BITS 08-15                  0174.000
                        **                          (USED BY SYSTEM RESET)                          0175.000
000000000000000F        TRACE       $EQ     15      CONTENTS :                                      0176.000
                        **                            BIT0=HALT                                     0177.000
                        **                            BIT1=STEP                                     0178.000
                        **                            BIT2=WAIT                                     0179.000
                        **                            BIT3=TRAP                                     0180.000
                        **                            BIT4=PRIV VIOLATION                           0181.000
                        **                            BIT5=UNDEF INSTR                              0182.000
                        **                            BIT6=1ST ACCESS TO IPL DVC                    0183.000
                        **                            BIT7=SAVED NPM                                0184.000
                        **                            BIT16=HALFWORD IND                            0185.000
                        **                            BIT17=INTERRUPTS BLOCKED                      0186.000
                        **                            BIT19=POWER FAIL                              0187.000
                        **                            BIT20=EXT IO RETURN FLAG                      0188.000
                        **                            BIT21=75 TRAP MODE FLAG                       0189.000
                        **                            BIT23=TRAP COND INSTR SKIP                    0190.000
                        **                            BIT28=ENTRY FROM EXTG                         0191.000
                        **                            BIT29=ENTRY FROM LATERR                       0192.000
                        **                            BIT30=ENTRY FROM EXTL                         0193.000
                        **                            BIT31=ENTRY FROM OTHERS                       0194.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC TSMAB+DR YYPCH+BDM+BAD ADDR

```
                                                                                                    0195.000
                    ************************************************************************        0196.000
                    **                                                                    **        0197.000
                    **                        UPPER BANK                                  **        0198.000
                    **                                                                    **        0199.000
                    ************************************************************************        0200.000
0000000000000000    TMP0            $EQU    0                                                       0201.000
0000000000000000    IOCD.ADDR       $EQU    0       IOCD MEMORY ADDRESS                             0202.000
0000000000000000    N.PSW2          $EQU    0       NEW PSW WORD 0                                  0203.000
                    **                                                                              0204.000
0000000000000001    TMP1            $EQU    1                                                       0205.000
0000000000000001    TD84            $EQU    1       TD 8000 & 4000 STATUS                           0206.000
0000000000000001    D.ADDR          $EQU    1       TRANSFER DATA ADDRESS (IOCD WORD 2)             0207.000
                    **                                                                              0208.000
0000000000000002    TMP2            $EQU    2                                                       0209.000
0000000000000002    TD2             $EQU    2       TD 2000 STATUS                                  0210.000
0000000000000002    CMD             $EQU    2       DEVICE COMMAND(CD 16-31 OR IOCD CMD 0-7)        0211.000
0000000000000002    B.IOCD2         $EQU    2       BINARY READ - IOCD WORD 2                       0212.000
0000000000000002    CPSTS           $EQU    2       CPU PSW STATUS                                  0213.000
0000000000000002    PSW1            $EQU    2       PSW WORD 1  (USED BY BRI AND LPSD)              0214.000
                    **                      CONTENTS                                                0215.000
                    **                              BIT0 = PRIV BIT                                 0216.000
                    **                              BITS 1-4 = CONDITION CODES 1-4                  0217.000
                    **                              BIT 5 = EXTENDED ADDRESS MODE                   0218.000
                    **                              BIT 6=LAST INSTRUCTION WAS IN RIGHT HALF        0219.000
                    **                                    WORD (55 MODE ONLY)                       0220.000
                    **                              BIT 6=ENABLE ARITHMETIC EXCEPTION TRAP          0221.000
                    **                                    (75 MODE ONLY)                            0222.000
                    **                                                                              0223.000
0000000000000003    TMP3            $EQU    3                                                       0224.000
0000000000000003    TCW             $EQU    3       TRANSFER CONTROL WORD OR TRANSFER COUNT          0225.000
0000000000000003    STATUS.CC       $EQU    3       EXTENDED I/O STATUS WORD RIGHT SHIFTED          0226.000
                    *                               FOR SETCC(5).  CC'S = BITS 1-4.                 0227.000
                    **                                                                              0228.000
0000000000000004    TMP4            $EQU    4                                                       0229.000
0000000000000004    IOCD1           $EQU    4       IOCD WORD 1                                     0230.000
0000000000000004    PSD.ADDR        $EQU    4       PROGRAM STATUS DOUBLE WORD ADDRESS              0231.000
                    *                               (USED BY BRI)                                   0232.000
0000000000000004    MSDLP           $EQU    4       CURRENT MAP SEGMENT DESCRIPTOR                  0233.000
                    **                              LIST POINTER.                                   0234.000
                    **                                CONTENTS :                                    0235.000
                    **                                  BITS 02/07  SEGMENT DESCRIPTOR              0236.000
                    **                                              COUNT.                          0237.000
                    **                                  BITS 08/31  MAP SEGMENT DESCRIPTOR          0238.000
                    **                                              LIST POINTER.                   0239.000
                    **                                                                              0240.000
                    **                                                                              0241.000
0000000000000005    TMP5            $EQU    5                                                       0242.000
0000000000000005    BRI.INDR        $EQU    5       BRI INDIRECT FLAGS IN BITS 0-7                  0243.000
                    **                                                                              0244.000
0000000000000006    TMP6            $EQU    6                                                       0245.000
0000000000000006    TI              $EQU    6       TCW DEDICATED MEMORY ADDRESS                    0246.000
0000000000000006    SI.VECTOR       $EQU    6       SI VECTOR (CONTENTS OF SI VECTOR ADDR)          0247.000
                    **                                                                              0248.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU C75R3
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
0000000000000007                TMP7        $EQU    7
0000000000000007                BUSREQ      $EQU    7       TEMP STORAGE FOR SEL BUS TRANSFER CODE
0000000000000007                TBL1        $EQU    7
                                **
0000000000000008                FDEV        $EQU    8       CHANNEL OR IOM ADDRESS (BITS 9-15)
0000000000000008                ICL.IOCD2   $EQU    8       ICL READ - IOCD WORD 2
0000000000000008                RZ          $EQU    8       USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS REGISTER R(Z) ADDRESS IN
                                *                           BITS 04-07.
0000000000000008                TBL2        $EQU    8
                                **
0000000000000009                RDEV        $EQU    9       I/O DEVICE OR RTOM INTERRUPT
0000000000000009                LOG.ADDR    $EQU    9       USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS LOGICAL ADDRESS FROM
                                *                           THE INSTRUCTION REGISTER R(Z).
0000000000000009                TBL3        $EQU    9
                                **
000000000000000A                INTR        $EQU    10      INTERRUPT LEVEL (BITS 9-15)
000000000000000A                LOG.MAP     $EQU    10      USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS THE MAP REGISTER ADDRESS
                                *                           IN BITS 03-07.
000000000000000A                TBL4        $EQU    10
                                **
000000000000000B                INTRTAB     $EQU    11      SCRATCH PAD INTERRUPT ENTRY
000000000000000B                ACCUM.CNT   $EQU    11      USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS THE ACCUMULATED MAP
                                *                           REGISTER COUNT IN BITS 00-07.
000000000000000B                TBL5        $EQU    11
                                **
000000000000000C                OFFSET      $EQU    12      SCRATCH PAD INTERRUPT ENTRY (WORKING)
000000000000000C                B.IOCD1     $EQU    12      BINARY READ - IOCD WORD 1
000000000000000C                PHY         $EQU    12      USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS THE COMPUTED PHYSICAL ADDRESS
000000000000000C                TBL6        $EQU    12
                                **
000000000000000D                INTRLOC     $EQU    13      SCRATCH PAD INTERRUPT ENTRY ADDRESS
000000000000000D                MID         $EQU    13      USED BY LEM, SEM, & CEMA INST.
                                *                           CONTAINS THE MAP IMAGE DESCRIPTOR
                                *                           HALFWORD IN BITS 16-31.
000000000000000D                TBL7        $EQU    13
                                **
000000000000000E                ICL.IOCD1   $EQU    14      ICL READ - IOCD WORD 1
000000000000000E                PNL.WRK     $EQU    14      SERIAL PANEL WORK REGISTER
000000000000000E                TBL8        $EQU    14
000000000000000E                MSD         $EQU    14      MAP SEGMENT DESCRIPTOR
                                *                             CONTENTS :
                                *                               BITS 02/07  SEGMENT PAGE COUNT
                                *                               BITS 08/31  MAP IMAGE DESCRIPTOR
                                *                                           LIST POINTER
                                **
000000000000000F                ZERO        $EQU    15      CONTENTS = 00000000
000000000000000F                SCR.MASK    $EQU    15      SCRATCH PAD CLEAR 'AND' MASK
                                **                          (USED BY SYSTEM RESET AND IPL)
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU SCRATCH PAD LAYOUT
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                0302.000
**          *************************                                           0303.000
**          *SCRATCH PAD ALLOCATION*                                           0304.000
**          *************************                                           0305.000
**               00 THRU 7F DEVICE ENTRIES                                      0306.000
**               80 THRU 81 BYTES 0-1 INTERRUPT ENTRIES                         0307.000
**                          BYTES 2-3 SI VECTOR ADDRESS                         0308.000
**               82 THRU 90 EMULATION WORK AREA & SERIAL PANEL CONTROL           0309.000
**               91         BYTES 0-3 CPU STATUS WORD                           0310.000
**               92 THRU FF BYTES 0-1 INTERRUPT ENTRIES                         0311.000
**                          BYTES 2-3 SI VECTOR ADDRESS                         0312.000
*****************************************************************             0313.000
*  00    *FLAGS*CLASS * INTR LEVEL  * IOC ADDR    * SUB ADDR    *              0314.000
*****************************************************************             0315.000
*  01    *FLAGS*CLASS * INTR LEVEL  * IOC ADDR    * SUB ADDR    *              0316.000
*****************************************************************             0317.000
*  02    *FLAGS*CLASS * INTR LEVEL  * IOC ADDR    * SUB ADDR    *              0318.000
*****************************************************************             0319.000
* THRU   *FLAGS*CLASS * INTR LEVEL  * IOC ADDR    * SUB ADDR    *              0320.000
*****************************************************************             0321.000
*  7F    *FLAGS*CLASS * INTR LEVEL  * IOC ADDR    * SUB ADDR    *              0322.000
*****************************************************************             0323.000
*  80    * INTR FLAGS * PTOM ENTRY  *     SI VECTOR ADDRESS      *              0324.000
*****************************************************************             0325.000
*  81    * INTR FLAGS * PTOM ENTRY  *     SI VECTOR ADDRESS      *              0326.000
*****************************************************************             0327.000
*  82    *   IOLD DEDICATED MAIN MEMORY ADDRESS  [ =700 ]        *              0328.000
*****************************************************************             0329.000
*  83    *   MASTER PROCESS LIST BASE ADDRESS [=784]             *              0330.000
*****************************************************************             0331.000
*  84    *   ICL/IPL DEVICE ENTRY (SEE 00/7F FOR FORMAT)         *              0332.000
*****************************************************************             0333.000
*  85    *   ACTIVATED INTERRUPT LEVEL COUNTER                   *              0334.000
*****************************************************************             0335.000
*  86    *   SERIAL PANEL CONTROL                                *              0336.000
*****************************************************************             0337.000
*  87    *   SERAIL PANEL TRANSMIT DATA                          *              0338.000
*****************************************************************             0339.000
*  88    *   SERIAL PANEL DISPLAY 'C'                            *              0340.000
*****************************************************************             0341.000
*  89    *   SERIAL PANEL DISPLAY 'A'                            *              0342.000
*****************************************************************             0343.000
*  8A    *   SERIAL PANEL DISPLAY 'D'                            *              0344.000
*****************************************************************             0345.000
*  8B    *   SERIAL PANEL DISPLAY 'B'                            *              0346.000
*****************************************************************             0347.000
*  8C    *   SERAIL PANEL DISPLAY FUNCTION                       *              0348.000
*****************************************************************             0349.000
*  8D    *   SERIAL PANEL INST, OPRND WRITE, OR OPRND READ STOP  *              0350.000
*****************************************************************             0351.000
*  8E    *   NOT USED                                            *              0352.000
*****************************************************************             0353.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU SCRATCH PAD LAYOUT
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                0354.000
*****************************************************************               0355.000
*  8F     *   IDENTIFY DEVICE PROTOCOL RESPONSE (DRT)            *               0356.000
*         *     (ONLY VALID, FOLLOWING A TPL)                    *               0357.000
*****************************************************************               0358.000
*  90     * CURRENT PSW WORD 2                                   *               0359.000
*****************************************************************               0360.000
*  91     * CPU STATUS WORD                                      *               0361.000
*****************************************************************               0362.000
*  92     * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS    *               0363.000
*****************************************************************               0364.000
*  93     * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS    *               0365.000
*****************************************************************               0366.000
*  94     * INTR FLAGS * DEV VECTOR  *      SI VECTOR ADDRESS    *               0367.000
*****************************************************************               0368.000
* THRU    * INTR FLAGS * DEV VECTOR  *      SI VECTOR ADDRESS    *               0369.000
*****************************************************************               0370.000
*  A3     * INTR FLAGS * DEV VECTOR  *      SI VECTOR ADDRESS    *               0371.000
*****************************************************************               0372.000
*  A4     * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS    *               0373.000
*****************************************************************               0374.000
* THRU    * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS    *               0375.000
*****************************************************************               0376.000
*  FF     * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS    *               0377.000
*****************************************************************               0378.000
**                                                                              0379.000
**  DEVICE ENTRY BREAKDOWN :                                                    0380.000
**                                                                              0381.000
**       ***********************************************************            0382.000
**       *FLAGS*CLASS * INTR LEVEL  * TOC ADDR    * SUB ADDR    *               0383.000
**       ***********************************************************            0384.000
**                                                                              0385.000
**       WHERE:                                                                 0386.000
**          FLAGS                                                               0387.000
**            BIT 0 = PROGRAM VIOLATION (NON-CLASS F ONLY)                      0388.000
**                1 = U/A                                                       0389.000
**                2 = U/A                                                       0390.000
**                3 = U/A                                                       0391.000
**          CLASS                                                               0392.000
**            BITS 4-7 = DEVICE TYPE                                            0393.000
**                0 = LINE PRINTER                                              0394.000
**                1 = CARD READER                                               0395.000
**                2 = TELETYPE                                                  0396.000
**                3 = RTOM-INTERVAL TIMER                                       0397.000
**                4-D = UNASSIGNED                                              0398.000
**                E = ALL I/O DEVICES EXCEPT TLC & CLASS F DEVICES              0399.000
**                F = ALL EXTENDED I/O DEVICES                                  0400.000
**          INTR LEVEL                                                          0401.000
**            INTERRUPT LEVEL ASSIGNED TO THIS DEVICE                           0402.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU SCRATCH PAD LAYOUT
PC TSMAR+DR YX PCH+BDM+BAD ADDR

```
                                                                                0403.000
**          IOC ADDR                                                            0404.000
**            PHYSICAL CONTROLLER ADDRESS                                       0405.000
**          SUB ADDR                                                            0406.000
**            SUB ADDRESS OF DEVICE                                             0407.000
**                                                                              0408.000
**                                                                              0409.000
**  INTERRUPT ENTRY BREAKDOWN :                                                 0410.000
**                                                                              0411.000
**      ********************************************************                0412.000
**      * INTR FLAGS * DEV VECTOR  *      SI VECTOR ADDRESS     *                0413.000
**      ********************************************************                0414.000
**      * INTR FLAGS * RTOM ENTRY  *      SI VECTOR ADDRESS     *                0415.000
**      ********************************************************                0416.000
**                                                                              0417.000
**          WHERE:                                                              0418.000
**            INTR FLAGS                                                        0419.000
**              BIT 0 = RAM LOADED                                              0420.000
**                  1 = I/O IN PROGRESS OR ECWCS ISSUED (CLASS F ONLY)          0421.000
**                  2 = CALM RBFLAG (LEVEL 27 ONLY)                             0422.000
**                  3 = NOT USED                                                0423.000
**                  4 = NOT USED                                                0424.000
**                  5 = ACTIVATE/DEACTIVATE INTERRUPT ISSUED                    0425.000
**                      0 = DEACTIVATE                                          0426.000
**                      1 = ACTIVATE                                            0427.000
**                  6 = REQUEST INTERRUPT ISSUED                                0428.000
**                  7 = ENABLE/DISABLE INTERRUPT ISSUED                         0429.000
**                      0 = DISABLE                                             0430.000
**                      1 = ENABLE                                              0431.000
**            DEV VECTOR                                                        0432.000
**              POINTER TO FIRST IOC DEV ENTRY ASSOCIATED WITH                  0433.000
**              THIS INTERRUPT LEVEL                                            0434.000
**            RTOM ENTRY                                                        0435.000
**              BIT 0 = 1 IF THIS ENTRY IS RTOM                                 0436.000
**              BITS 1-3 = LSB'S OF PHYSICAL RTOM BOARD NUMBER                  0437.000
**              BITS 4-7 = LSB'S OF RTOM SUB ADDRESS                            0438.000
**            SI VECTOR ADDRESS                                                 0439.000
**              BITS 0-15 CONTAIN THE DEDICATED MEMORY ADDRESS FOR              0440.000
**              THE INTERRUPT LEVEL VECTOR.                                     0441.000
**                                                                              0442.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU SCRATCH PAD LAYOUT
PC TSMAB+DM YX PCH+BDM+RAD ADDR

```
                                                                                0443.000
**                                                                              0444.000
**  PSW WORD 2 BREAK DOWN :                                                     0445.000
**                                                                              0446.000
*********************************************************************           0447.000
*  90    * CURRENT PSW WORD 2       *              *              *             0448.000
*********************************************************************           0449.000
**                                                                  *           0450.000
**       WHERE :                                                    *           0451.000
**       BITS                                                       *           0452.000
**            0-1    MAP GRANULARITY AS FOLLOWS :                   *           0453.000
**                   0 0  OPERATE UNMAPPED                          *           0454.000
**                   0 1  OPERATE WITH 8K WORD GRANULARITY          *           0455.000
**                   1 0  OPERATE WITH 8K WORD GRANULARITY          *           0456.000
**                   1 1  OPERATE WITH 8K WORD GRANULARITY          *           0457.000
**                                                                  *           0458.000
**            2-13   BORROW PROCESS INDEX (BPIX)                    *           0459.000
**            14     NOT USED                                       *           0460.000
**                                                                  *           0461.000
**            15     RETAIN CURRENT MAP                             *           0462.000
**                                                                  *           0463.000
**            16-17  EXTERNAL INTERRUPT FLAGS AS FOLLOWS :          *           0464.000
**                   0 0  OPERATE WITH INTERRUPT LEVEL ACTIVE       *           0465.000
**                   0 1  OPERATE WITH INTERRUPTS BLOCKED           *           0466.000
**                   1 0  RETAIN CURRENT BLOCKING MODE              *           0467.000
**                   1 1  RETAIN CURRENT BLOCKING MODE              *           0468.000
**                                                                  *           0469.000
**            18-29  CONTROL PROCESS INDEX (CPIX)                   *           0470.000
**            30-31  NOT USED                                       *           0471.000
**                                                                  *           0472.000
*********************************************************************           0473.000
```

```
                                                                        0474.000
**                                                                      0475.000
**      CPU/IPU STATUS WORD BREAKDOWN:                                  0476.000
**                                                                      0477.000
************************************************************************0478.000
*   91    *          CPU STATUS WORD                                   *0479.000
************************************************************************0480.000
**        WHERE:                                                        *0481.000
**        BIT 0      =0,  CLASS 0,1,2, OR F ERROR                       *0482.000
**                   =1,  CLASS F (EXTENDED I/O) ERROR                  *0483.000
**             1     =0, I/O PROCESSING ERROR                           *0484.000
**                   =1, INTERRUPT PROCESSING ERROR                     *0485.000
**             2     FINAL BUS TRANSFER ERROR                           *0486.000
**             3     BUS NO RESPONSE ERROR                              *0487.000
**             4     I/O CHANNEL BUSY OR BUSY STATUS BIT ERROR          *0488.000
**             5     READY TIMEOUT ERROR                                *0489.000
**             6     I/O DAT TIMEOUT ERROR                              *0490.000
**             7     RETRY COUNT EXHAUSTED ERROR                        *0491.000
**             8     OPERAND FETCH PARITY ERROR                         *0492.000
**             9     INSTRUCTION FETCH PARITY ERROR                     *0493.000
**             10    OPERAND NON-PRESENT ERROR                          *0494.000
**             11    INSTRUCTION NON-PRESENT ERROR                      *0495.000
**             12    UNDEFINED 75 MODE INSTRUCTION ERROR                *0496.000
**             13    MEMORY FETCH DAT TIMEOUT ERROR                     *0497.000
**             14    RESET CHANNEL ERROR                                *0498.000
**             15    CHANNEL WCS NOT ENABLED ERROR                      *0499.000
**             16    MAP NOT FOUND (LEM,SEM,CEMA INST. ONLY)            *0500.000
**                   OR MAP REGISTER ADDRESS OVERFLOW (MAP              *0501.000
**                   CONTEXT SWITCH).                                   *0502.000
**             17    UNEXPLAINED MEMORY ERROR                           *0503.000
**             18    NPI I/O ERROR                                      *0504.000
**             19    UNDEFINED INSTRUCTION 55 MODE ONLY                 *0505.000
**             20    MAP INVALID ACCESS OR MAP MODE RESTRICTION         *0506.000
**                   ERROR.                                             *0507.000
**             21    IPL I/O OR MEMORY ERROR FLAG                       *0508.000
**                   /IPU PRIVILEGE VIOLATION                           0509.000
**             22    IPU POWER FAIL OCCURED                             *0510.000
**             23    IPU ARITHMETIC EXCEPTION                           *0511.000
**             24    ENABLE ARITHMETIC EXCEPTION TRAP                   *0512.000
**             25    DISABLE 75 MODE TRAPS                              *0513.000
**             26    BLOCK MODE IS ACTIVE                               *0514.000
**             27    CPU POWER FAIL UP MEMORY ERROR                     *0515.000
**                   /IPU MODE STATUS BIT                               0516.000
**             28    NOT USED                                           *0517.000
**             29    CPU ELSA MODE                                      *0518.000
**             30    NOT USED                                           *0519.000
**             31    =0, CPU MODE 55                                    *0520.000
**                   =1, CPU MODE 75                                    *0521.000
************************************************************************0522.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/TPU SCRATCH PAD LAYOUT
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                                                    0523.000
                        ***                                                                                         0524.000
                        ***  CPU STATUS WORD ERROR FLAG EQUATES                                                      0525.000
                        ***                                                                                         0526.000
80000000                EXT.TO.FLG      $EQ     @80000000  CLASS F (EXTENDED I/O) ERROR FLAG                       0527.000
40000000                INT.ERR.FLG     $EQ     @40000000  INTERRUPT SEQUENCE ERROR FLAG                          0528.000
20000000                FINAL.FLG       $EQ     @20000000  FINAL TRANSFER ERROR FLAG                              0529.000
10000000                N.RESP.FLG      $EQ     @10000000  I/O NO RESPONSE ERROR FLAG                            0530.000
08000000                BUSY.FLG        $EQ     @08000000  I/O CHANNEL BUSY OR BUSY STATUS                       0531.000
                        *                                  ERROR FLAG                                           0532.000
04000000                RDY.TIM.FLG     $EQ     @04000000  READY TIMEOUT ERROR FLAG                             0533.000
02000000                IO.TIM.FLG      $EQ     @02000000  I/O DRT TIMEOUT ERROR FLAG                           0534.000
01000000                CNT.EXH.FLG     $EQ     @01000000  RETRY COUNT EXHAUSTED ERROR FLAG                     0535.000
00800000                OP.PE.FLG       $EQ     @00800000  OPERAND FETCH PARITY ERROR FLAG                     0536.000
00400000                INST.PE.FLG     $EQ     @00400000  INST.  FETCH PARITY ERROR FLAG                      0537.000
00200000                OP.NPM.FLG      $EQ     @00200000  OPERAND NON-PRESENT ERROR FLAG                      0538.000
00100000                INST.NPM.FLG    $EQ     @00100000  INST. NON-PRESENT MEMORY ERROR FLAG                 0539.000
00060000                UNDEF.75.FLG    $EQ     @00060000  UNDEFINED INSTRUCTION 75 MODE ONLY                  0540.000
00040000                TIM.FLG         $EQ     @00040000  MEMORY FETCH DRT TIMEOUT ERROR FLAG                 0541.000
00020000                RSTCHL.FLG      $EQ     @00020000  RESET CHANNEL ERROR FLAG                            0542.000
00010000                CHAN.WCS.N.FLG  $EQ     @00010000  CHANNEL WCS NOT ENABLED FLAG                        0543.000
00008000                MAP.N.FOUND.FLG $EQ     @00008000  MAP NOT FOUND (LEM, SEM, CEMA INST)                 0544.000
                        *                                  OR MAP REGISTER ADDRESS OVERFLOW                     0545.000
                        *                                  (MAP CONTEXT SWITCH).                                0546.000
00004000                UNEXP.M.ERR.FLG $EQ     @00004000  UNEXPLAINED MEMORY ERROR                            0547.000
00002000                DRT.ERR.FLG     $EQ     @00002000  DRT I/O ERROR FLAG                                  0548.000
00001000                UNDEF.55.FLG    $EQ     @00001000  UNDEFINED INSTRUCTION 55 MODE ONLY                  0549.000
00000800                MAP.INVALID.FLG $EQ     @00000800  MAP INVALID ACCESS                                  0550.000
00000400                IPL.ERR.FLG     $EQ     @00000400  IPL OR I/O MEMORY ERROR FLAG                        0551.000
00000010                PWR.F.UP.FLG    $EQ     @00000010  POWER FAIL UP MEMORY ERROR FLAG                     0552.000

                                                                                                                    0553.000
                        ***                                                                                         0554.000
                        ***  EXTENDED I/O STATUS FLAGS AND CC'S                                                     0555.000
                        ***                                                                                         0556.000
80000000                SCT.FLG         $EQ     @80000000 SYSTEM CHECK TRAP ROUTING FLAG                       0557.000
                        *                                 (ALSO PROVIDES A EXT I/O CPU STATUS FLAG)             0558.000
10000000                CC3             $EQ     @10000000 S REG CONDITION CODE 3 FLAG                          0559.000
08000000                CC4             $EQ     @08000000 S REG CONDITION CODE 4 FLAG                          0560.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75A CPU/IPU SCRATCH PAD LAYOUT
PC TSMAB+DR YYPCH+RDM+BAD ADDR

```
                                                                        0561.000
*************************************************************************0562.000
**                  EXCEPTIONAL CONDITION GROUPINGS                   **0563.000
*************************************************************************0564.000
**                                                                      0565.000
**              EXTERNAL EVENT UNIVERSAL                                0566.000
**                      EXTERNAL EVENT GLOBAL                           0567.000
**                      NOT ENABLE INTERRUPT FF (FF7)                   0568.000
**                                                                      0569.000
**              EXTERNAL EVENT GLOBAL                                   0570.000
**                      EXTERNAL EVENT LOCAL                            0571.000
**                      PANEL ATTENTION                                 0572.000
**                      TRACEFF(FF3)                                    0573.000
**                      INSTRUCTION TIMEOUT                             0574.000
**                      UART XMTT BUF EMPTY (UARTTBMT)                  0575.000
**                      UART DATA AVAILABLE (UARTDAV)                   0576.000
**                      IPU.START (IPU CAUSING TRAP TO THE CPU)         0577.000
**                                                                      0578.000
**              LATE ERRORS                                             0579.000
**                      MUX INSTRUCTION ERROR (PARITY ERROR)            0580.000
**                      MUX OPERAND ERROR (PARITY ERROR)                0581.000
**                      ADDRESS STOP                                    0582.000
**                      INSTRUCTION NORESPONSE (NON-PRESENT MEM)        0583.000
**                      ARITHMETIC EXCEPTION                            0584.000
**                      BLOCK MODE TIMEOUT ERROR                        0585.000
**                                                                      0586.000
**              EXTERNAL EVENT LOCAL                                    0587.000
**                      OPERAND TIMEOUT                                 0588.000
**                      OPERAND NORESPONSE (NON-PRESENT MEMORY)         0589.000
**                      OPERAND PE (PARITY ERROR)                       0590.000
**                      NOT RUNFF  (CPU HALTED)                         0591.000
**                      FFINT  (EXTERNAL INTERRUPT)                     0592.000
**                      SYSTEM RESET F/F                                0593.000
**                      POWER FAIL                                      0594.000
**                      IPL                                             0595.000
**                      USER PANEL ATTENTION                            0596.000
**                      PROTECT VIOLATION                               0597.000
**                      MAP INVALID ACCESS                              0598.000
**                      IPU.START (IPU CAUSING TRAP TO THE CPU)         0599.000
**                                                                      0600.000
**              EXTERNAL EVENT FLIP/FLOPS                               0601.000
**                      FF0(REG BANK SEL)                               0602.000
**                      FF1(EXECUTE FLAG)                               0603.000
**                      FF2(PRIVILEGE BIT)                              0604.000
**                              0 = PRIVILEGED                          0605.000
**                              1 = NON-PRIVILEGED                      0606.000
**                      FF3(TRACE FLAG)                                 0607.000
**                      FF4(DISPLAY PARITY ERROR)                       0608.000
**                      FF5(DISPLAY INTERRUPT ACTIVE)                   0609.000
**                      FF6(DISPLAY WAIT MODE)                          0610.000
**                      FF7-ENABLE INTERRUPT FF                         0611.000
```

```
                                                                                0612.000
**          FIRST RTOM BOARD ASSIGNMENTS :                                      0613.000
**                                                                              0614.000
**                        SUB ADDR        INTERRUPT LEVEL-TYPE                  0615.000
**                        0E              00      POWER FAIL                    0616.000
**                        0E              01      SYSTEM OVERRIDE               0617.000
**                        0D              12      MEMORY PARITY                 0618.000
**                        0C              13      CONSOLE INTERRUPT             0619.000
**                        0B              24      NONPRESENT MEMORY             0620.000
**                        0A              25      UNDEFINED INSTRUCT            0621.000
**                        09              26      PRIVILEGE VIOLAT              0622.000
**                        08              27      CALM                          0623.000
**                        07              29      ARITHMETIC EXCEP              0624.000
**                        06              28      REAL TIME CLOCK               0625.000
**                        05              2A      EXTERNAL #0                   0626.000
**                        04              2B      EXTERNAL #1                   0627.000
**                        03              2C      EXTERNAL #2                   0628.000
**                        02              2D      EXTERNAL #3                   0629.000
**                        01              2E      EXTERNAL #4                   0630.000
**                        00              2F      EXTERNAL #5                   0631.000
          $USE  DEF.75H                                                         0632.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU EXTERNAL EVENT HANDLER
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    0633.000
                                                    **                                                              0634.000
                                                    **      ENTRY TO THIS FLOW IS BY A JUMPZ OR MASTER CLEAR        0635.000
                                                    **                                                              0636.000
                                                                                                                    0637.000
                                                    DECODE                                                          0638.000
0000 7 2 6 0 1 0 F 0 0C 0 104 0 0 0 0 0 0 0 0               RESET(EXEF),RSTRHF,DECODE(#),T=MAR,IF NOEXTUNIV *JUMPD;  0639.000
                                                                                                                    0640.000
                                                    **                                                              0641.000
                                                    **      ENTRY TO THIS FLOW IS DUE TO A EXTERNAL GLOBAL OR       0642.000
                                                    **        EXTERNAL UNIVERSAL CONDITION IN THE RUN MODE          0643.000
                                                    **                                                              0644.000
                                                    **      NOTE:  IF EXT UNIVERSAL =1, AND EXTERNAL GLOBAL =0, ENTRY WAS 0645.000
                                                    **               CAUSED BY THE RESET CONDITION OF THE ENABLE    0646.000
                                                    **               INTERRUPT FLIP/FLOP.                           0647.000
                                                    **                                                              0648.000
                                                                                                                    0649.000
                                                    EXTG                                                            0650.000
0001 9 4 7 0 1 0 0 0 00 0 EEC 0 0 0 0 0 0 0 0 0   C         IF %EXTG *LINK TEST.NON.INTERRUPTABLE;                  0651.000
0002 6 0 0 3 4 A 6 7 02 0 087 0 0 0 0 0 0 0 0               FR(TRACE)=#00000008:FR(TRACE),CLRS;  INDICATE ENTRY FROM EXTG 0652.000
                                                                                                                    0653.000
                                                    **                                                              0654.000
                                                    **      ENTRY TO THIS FLOW IS DUE TO A EXTERNAL UNIVERSAL, EXTERNAL 0655.000
                                                    **      GLOBAL, EXTERNAL LOCAL, OR LATE ERROR IN THE RUN MODE   0656.000
                                                    **                                                              0657.000
                                                                                                                    0658.000
                                                    EXTPROC                                                         0659.000
0003 0 6 7 0 1 0 0 0 00 0 117 0 0 0 0 0 0 0 0 0 117         *LINK BACKDATE.PC;                                      0660.000
                                                                                                                    0661.000
                                                    **                                                              0662.000
                                                    **      THIS FLOW PROVIDES THE BASIC IDLE LOOP (NOT RUN MODE).  0663.000
                                                    **                                                              0664.000
                                                    **      ENTRY IS CAUSED BY A EXTERNAL UNIVERSAL, EXTERNAL GLOBAL, 0665.000
                                                    **        EXTERNAL LOCAL, OR LATE ERROR CONDITION IN THE RUN MODE 0666.000
                                                    **        OR NORMAL EVENT SCHEDULING IN THE IDLE LOOP.          0667.000
                                                    **                                                              0668.000
                                                                                                                    0669.000
                                                    EVENT.POLL                                                      0670.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU EXTERNAL EVENT HANDLER

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0004 1 4 0 0 1 0 0 0 00 0 AE6 0 0 0 0 0 0 0 0   6        IF LATERR:EXTG *HOP $+2;                                          0671.000
0005 0 5 0 0 5 0 0 0 00 0 F3F 0 0 0 0 0 0 0 0  3F        NOD=R(TRACE,HWS), *GO TO TEST.FFINT;GO TO TEST FOR INT            0672.000
0006 9 4 0 0 1 0 0 0 00 8 B98 0 0 0 0 0 0 0 0   8        PCTOMAR, IF %PWRFAIL *GO TO $+2;                                  0673.000
0007 6 5 0 0 1 0 B 6 00 0 0FE 0 0 0 0 0 0 0 0  FE        FR(TEMP4)=MAR,   *GOTO POWER.FAIL; SAVE PC IN TEMP4               0674.000
0008 9 4 0 0 1 0 0 0 00 0 F8A 0 0 0 0 0 0 0 0   A        IF %TRACE *HOP $+2; TEST NOT ALREADY HALTED                       0675.000
0009 3 5 7 0 4 0 F 0 00 3 FDA 0 0 0 0 0 0 0 0  DA        T=R(TRACE), IF FFRUN *LINK TRACE.RUN;                             0676.000
000A 5 5 0 0 4 0 F 0 00 3 FBF 0 0 0 0 0 0 0 0  BF        T=R(TRACE), IF %FFRUN *GO TO STOP;                                0677.000
000B 0 5 0 0 5 0 F 0 00 0 F10 0 0 0 0 0 0 0 0  10        T=R(TRACE,HWS), *GO TO MISC.INTERRUPTS;                           0678.000

                                                                                                                           0679.000
                                                 TEST.NON.INTERRUPTABLE                                                    0680.000
000C 0 0 0 0 1 0 0 0 00 0 F04 0 0 0 0 0 0 0 0            SET(ENAINTFF); ENABLE INTERRUPTS                                  0681.000
000D 0 0 4 0 1 0 F 0 00 0 0F7 0 0 0 0 0 0 0 0            T=MAR, DECODE(@F), CLRS; SET UP PRIME DECODE OF IO 00-05          0682.000
000E 7 2 6 0 1 0 0 0 00 0 104 0 0 0 0 0 0 0 0            DECODE(#), RESET(FXFF), IF NOFXTUNIV *JUMPD:EXIT TO PRIME DECODE  0683.000
                                                 DUD.0XX                                                                   0684.000
000F 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0            *JUMPJ;                                                           0685.000
                                                                                                                           0686.000
                                                 **  ENTRY PARAMETERS :                                                    0687.000
                                                 **    T=R(TRACE,HWS)                                                      0688.000
                                                 **                                                                        0689.000
                                                                                                                           0690.000
                                                 MISC.INTERRUPTS                                                           0691.000
0010 0 0 0 0 0 0 0 0 00 0 005 0 0 0 0 0 0 0 0            NOD=T, RESET(ENAUORD); TEST WAIT BIT                              0692.000
0011 5 5 0 0 1 0 0 0 00 A 0EF 0 0 0 0 0 0 0 0  FF        IF %RMUX18 *GO TO RENABLE.EXIT;  BRANCH IF WAIT BIT               0693.000
0012 6 0 0 3 4 D 0 7 02 0 040 0 0 0 0 0 0 0 0            NOD=@00000004&FR(TRACE);         TEST ENTRY FROM LATERR           0694.000
0013 6 0 0 3 4 D 0 7 02 0 010 0 0 0 0 0 0 0 0            NOD=@00000001&FR(TRACE);         TEST ENTRY FROM OTHERS           0695.000
0014 C 4 0 0 1 0 0 0 00 0 00B 0 0 0 0 0 0 0 0  1B        IF NALUZ *GO TO LATERR.TEST;                                      0696.000
0015 C 5 0 0 1 0 0 0 00 0 0E9 0 0 0 0 0 0 0 0  E9        IF NALUZ *GO TO OTHERS.TEST;                                      0697.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                    E X T E R N A L   E V E N T   H A N D L E R
PC   T S M A B + B R  Y X PCH + B D M + B A D  ADDR

0016 3 4 0 7 4 B 0 0 00 C FDB 0 0 0 0 0 0 0 0  1B        NOD=%@FFFFFFFD&R(TRACE), IF LATERRW *GO TO LATERR.TEST;                0698.000

0017 6 0 0 3 4 D 0 7 02 0 080 0 0 0 0 0 0 0 0            NOD=@00000000&FR(TRACE);        TEST ENTRY FROM EXTG                  0699.000

0018 C 5 0 0 1 0 0 0 00 0 033 0 0 0 0 0 0 0 0  33        IF NALUZ *GO TO EXTL.TEST;                                            0700.000

0019 B 5 0 0 1 0 0 0 00 0 03F 0 0 0 0 0 0 0 0  3F        IF ALUZ *GO TO TEST.FFINT;                                            0701.000

001A 0 5 0 0 1 0 0 0 00 0 026 0 0 0 0 0 0 0 0  26        *GO TO EXTG.TEST;                                                     0702.000

                                                                                                                               0703.000
                                                **              ENTRY FROM LATE ERROR                                          0704.000
                                                **                      MUX OPERAND ERROR       (PRIOR INSTR)          0705.000
                                                **                      ARITHMETIC EXCEPTION    (PRIOR INSTR)          0706.000
                                                **                      MUX INSTRUCTION ERROR   (CURRENT INSTR)        0707.000
                                                **                      INSTRUCTION NORESPONSE  (CURRENT INSTR)        0708.000
                                                **                                                                     0709.000
                                                                                                                               0710.000
                                                LATERR.TEST                                                                    0711.000

001B 9 4 0 0 1 0 0 0 00 0 FCD 0 0 0 0 0 0 0 0  1D        IF %INSTMIUER *GO TO $+2;       MUX INSTR PARITY ERROR                0712.000

001C 0 6 7 0 1 0 0 0 00 0 541 0 0 0 0 0 0 0 0  541       *LINK INST.PE;                                                        0713.000

001D 9 4 0 0 1 E E 0 02 0 FDF 0 0 0 0 0 0 0 0  1F        FULLMAR=@FD000000, IF %OPMIUER *GO TO $+2;  CLEAR MAR                 0714.000

001E 0 6 7 0 1 0 0 0 00 0 540 0 0 0 0 0 0 0 0  540       *LINK OPRND.PE;                 OPERAND PARITY ERROR                  0715.000

001F 9 4 0 0 1 0 0 0 00 0 BF1 0 0 0 0 0 0 0 0  21        IF %AEXP *GO TO $+2;            ARITHMETIC EXCEPTION                  0716.000

0020 0 6 7 0 1 0 0 0 00 0 27C 0 0 0 0 0 0 0 0  27C       *LINK ARITHMETIC.EXCEPTION;                                           0717.000

0021 9 4 0 0 1 0 0 0 00 0 7C4 0 0 0 0 0 0 0 0  24        IF %INSTNORESP *GO TO $+3;      INSTR NONPRESENT MEMORY               0718.000

0022 6 0 0 2 4 A B 7 02 0 010 0 0 0 0 0 0 0 0            FR(TRACE)=@00000100:FR(TRACE);  INDICATE INSTR SKIP                   0719.000

0023 0 6 7 0 1 0 0 0 00 0 548 0 0 0 0 0 0 0 0  548       *LINK INSTRUCTION.NONPRESENT;                                         0720.000

                                                CK.BLK.TIMEOUT                                                                 0721.000

0024 4 5 0 0 1 0 0 0 00 0 03F 0 0 0 0 0 0 0 0  3F        IF %BLKTIMEOUT *GO TO TEST.FFINT; EXIT IF NO LATE ERRORS FOUND        0722.000

0025 0 6 0 0 1 0 0 0 00 0 55C 0 0 0 0 0 0 0 0  55C       *GO TO BLOCK.MODE.TIMEOUT;  EXIT TO BLOCK TRAP                        0723.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU EXTERNAL EVENT HANDLER
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                0724.000
                                                                                                                                0726.000
                                        **                                                                                      0727.000
                                        **              ENTRY FROM GLOBAL SIGNALS                                               0728.000
                                        **                      OPERAND TIMEOUT         (PRIOR INSTR)                           0729.000
                                        **                      OPERAND NORESPONSE      (PRIOR INSTR)                           0730.000
                                        **                      PROTECT VIOLATION       (PRIOR INSTR)                           0731.000
                                        **                      INSTRUCTION TIMEOUT     (CURRENT INSTR)                         0732.000
                                        **                      USER PANEL ATTENTION    (CURRENT INSTR)                         0733.000
                                        **                      UART TRANSMIT BUFFER EMPTY                                      0734.000
                                        **                      UART DATA AVAILABLE                                             0735.000
                                        **                      IPU.START (IPU CAUSING TRAP TO CPU)                             0736.000
                                        **                                                                                      0737.000

                                        EXTS.TEST                                                                               0738.000

0026 9 4 0 0 1 E E 0 02 0 508 0 0 0 0 0 0 0 0   28      FULLMAR=w50000000, IF %OPNORESP:OPTIMEOUT *GO TO $+2;CLEAR MAR           0739.000

0027 0 6 7 0 1 0 0 0 00 0 547 0 0 0 0 0 0 0 0   547     *LINK OPERAND.NONPRESENT;                                               0740.000

0028 9 4 0 0 1 0 0 0 00 0 7EA 0 0 0 0 0 0 0 0   2A      IF %PROTV *GO TO $+2;                                                   0741.000

0029 0 6 7 0 1 0 0 0 00 0 551 0 0 0 0 0 0 0 0   551     *LINK PRIVILEGE.VIOLATION;                                              0742.000

002A 9 4 0 0 1 0 0 0 00 0 DCD 0 0 0 0 0 0 0 0   2D      IF %INSTTIMEOUT *GO TO $+3;                                             0743.000

002B 6 0 0 2 4 A B 7 02 0 010 0 0 0 0 0 0 0 0           FP(TRACE)=w00000100:FP(TRACE);  INDICATE INSTR SKIP                     0744.000

002C 0 6 7 0 1 0 0 0 00 0 54C 0 0 0 0 0 0 0 0   54C     *LINK INSTRUCTION.TIMEOUT;                                              0745.000

002D 9 4 0 0 1 0 0 0 00 0 BAF 0 0 0 0 0 0 0 0   2F      IF %UPREQ *GO TO $+2;                                                   0746.000

002E 0 6 7 0 1 0 0 0 00 0 279 0 0 0 0 0 0 0 0   279     *LINK CONSOLE.INTERRUPT;                                                0747.000

002F 4 6 0 0 1 0 0 0 00 3 F71 0 0 0 0 0 0 0 0   F71     IF IPU.START *GO TO EXTERNAL.IPU.TEST+2;                                0748.000

0030 9 4 0 0 1 0 0 0 00 0 D12 0 0 0 0 0 0 0 0   32      IF %UARTDAV:UARTTRMT *HOP $+2;                                          0749.000

0031 0 6 7 0 1 0 0 0 00 0 A5F 0 0 0 0 0 0 0 0   A5F     *LINK S.P.ATTN;                                                         0750.000

0032 0 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0   3F      *GO TO TEST.FFINT;                                                      0751.000
```

```
                                                              **            ENTRY FROM EXTL
                                                              **                    OPERAND TIMEOUT           (CURRENT INSTR)
                                                              **                    OPERAND NORESPONSE        (CURRENT INSTR)
                                                              **                    OPERAND PARITY ERROR      (CURRENT INSTR)
                                                              **                    POWER FAIL                (CURRENT INSTR)
                                                              **                    USER PANEL ATTENTION      (CURRENT INSTR)
                                                              **                    PROTECT VIOLATION         (CURRENT INSTR)
                                                              **                    IPU.START  (IPU CAUSING TRAP TO CPU)
                                                              **

                                                              EXTL.TEST

0033 9 4 0 0 1 0 0 0 00 0 5D6 0 0 0 0 0 0 0 0    36                 IF %OPNORESP:OPTIMEOUT *GO TO $+3;

                                                              EXTG.NPM.ENTRY

0034 6 0 0 2 4 A B 7 02 0 017 0 0 0 0 0 0 0 0                       FR(TRACE)=@00000100:FR(TRACE),CLRS;  INDICATE INSTR SKIP

0035 0 6 7 0 1 E E 0 00 0 547 0 0 0 0 0 0 0 0   547                  FULLMAR=S,*LINK OPERAND.NONPRESENT;

0036 9 4 0 0 1 0 0 0 00 0 F59 0 0 0 0 0 0 0 0    39                 IF %OPRNDPE *HOP $+3;

0037 6 0 0 2 4 A B 7 02 0 017 0 0 0 0 0 0 0 0                       FR(TRACE)=@00000100:FR(TRACE), CLRS: INDICATE INSTRUCTION SKIP

0038 0 6 7 0 1 0 0 0 00 0 540 0 0 0 0 0 0 0 0   540                  *LINK OPRND.PF; GO PROCESS PARITY ERROR IN CURRENT INSTRUCTION

0039 9 4 0 0 1 0 0 0 00 0 BAB 0 0 0 0 0 0 0 0    3B                 IF %UPRFG *GO TO $+2;            USER PANEL ATTENTION

003A 0 6 7 0 1 0 0 0 00 0 279 0 0 0 0 0 0 0 0   279                  *LINK CONSOLE.INTERRUPT;

003B 9 4 0 0 1 0 0 0 00 0 7EE 0 0 0 0 0 0 0 0    3E                 IF %PROTV *HOP $+3;

003C 6 0 0 2 4 A B 7 02 0 010 0 0 0 0 0 0 0 0                       FR(TRACE)=@00000100:FR(TRACE);  INDICATE INSTR SKIP

003D 0 6 7 0 1 0 0 0 00 0 551 0 0 0 0 0 0 0 0   551                  *LINK PRIVILEGE.VIOLATION;

003E 4 6 0 0 1 0 0 0 00 3 F71 0 0 0 0 0 0 0 0   F71                  IF IPU.START *GO TO EXTERNAL.IPU.TEST+2;
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                E X T E R N A L   E V E N T   H A N D L E R
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    0778.000

                                                                                                                    0779.000
                                                    **                                                              0780.000
                                                    **          FUNNEL--COMMON ENTRY FROM RAW SIGNAL SCHEDULING     0781.000
                                                    **                                                              0782.000

                                                                                                                    0783.000

                                                    TEST.FFINT                                                      0784.000

003F 0 0 0 0 4 0 F 0 00 0 900 0 0 0 0 0 0 0 0           T=R(STMASK);                                                0785.000

0040 0 0 0 5 0 0 F 0 00 0 F04 0 0 0 0 0 0 0 0           T=STATUS&T, SET(EMAINTFF); GET CPU STATUS                   0786.000

0041 0 0 0 5 6 8 1 0 02 0 804 0 0 0 0 0 0 0 0           S=%0FF80FFFF&INTLVL, SET(HIREG); GET POLLING INT LEVEL      0787.000

0042 1 4 0 0 1 E 0 0 00 0 DA7 0 0 0 0 0 0 0 0   47      NOD=S, IF FFINT *HOP INTERRUPT;                             0788.000

0043 0 5 0 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0   F0      *GO TO FILL.PIPELINE;                                       0789.000
```

```
            S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    I N T E R R U P T  S P B
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    0790.000

                                                                                                                    0791.000
                                                        **                                                          0792.000
                                                        **          EXTERNAL INTERRUPT PROCESSOR                    0793.000
                                                        **                                                          0794.000
                                                        **             T = CPU STATUS IN BITS 00-12                 0795.000
                                                        **             S = POLLING INTERRUPT LEVEL IN BITS 09-15    0796.000

                                                                                                                    0797.000
                                                        (@47)                                                       0798.000
                                                        INTERRUPT                                                   0799.000
0047 0 0 0 0 0 0 8 0 00 0 204 0 0 0 0 0 0 0 0                   R(CPSTS)=T, RESET(PRIV); SAVE CPU STATUS & ENTER PRIV MODE  0800.000
0048 4 4 0 0 1 E 8 0 00 8 A0A 0 0 0 0 0 0 0 0  4A              R(INTR)=S, IF UNBLOCK *GO TO $+2; SAVE INTERRUPT LEVEL      0801.000
0049 0 6 7 0 1 0 0 0 00 0 FE8 0 0 0 0 0 0 0 0  FE8             *LINK SET.BLOCKED.M.FLAG;                                  0802.000
004A 0 6 7 0 1 0 0 0 00 0 467 0 0 0 0 0 0 0 0  467             *LINK BUILD.INTR.ADDR;                                     0803.000
                                                                                                                    0804.000
                                                        **   RETURNED DATA                                          0805.000
                                                        **      R(INTR)     = INTERRUPT NUMBER                      0806.000
                                                        **      R(RDEV)     = FORMATED REAL ADDRESS                 0807.000
                                                        **      R(INTRTAB)  = INTERRUPT TAB ENTRY                   0808.000
                                                        **      R(SI.VECTOR) = SI VECTOR (CONTENTS OF VECTOR ADDR)  0809.000
                                                        **      R(N.PSW2) = NEW PSW WORD 2  (75 MODE ONLY)          0810.000
                                                                                                                    0811.000
004B 0 0 0 0 4 1 0 0 00 0 900 0 0 0 0 0 0 0 0                   NOD=%R(RDEV);  TEST FOR CLASS F DEVICES                    0812.000
004C 0 0 0 0 1 0 0 0 00 0 704 0 0 0 0 0 0 0 0                   RESET(ENAINTFF);          DISABLE INTERRUPTS               0813.000
004D E 4 0 0 4 0 E 1 00 0 70F 0 0 0 0 0 0 0 0  4F              FULLMAR=FR(RDEV), IF %ALU4-7Z *GO TO $+2; BRANCH NOT CLASS F  0814.000
004E 0 6 0 0 1 0 0 0 00 0 E68 0 0 0 0 0 0 0 0  E68             *GO TO DEV.CLASSF.INT;  GO HANDLE CLASS F INTERRUPT        0815.000
004F 0 0 0 0 5 0 0 0 00 D F00 0 0 0 0 0 0 0 0                   NOD=R(TRACE,HWS),OTHERBANK; TEST FOR TRAP INTERRUPT        0816.000
0050 3 4 0 3 1 E F 0 02 B 102 0 0 0 0 0 0 0 0  52              T=@00000010, IF BMUX19 *GOTO ACK.INTR; SET ACK INTR CODE   0817.000
0051 0 5 0 0 4 0 0 0 00 0 B87 0 0 0 0 0 0 0 0  87              NOD=R(INTRTAB), *GO TO TRAP; TEST RAM LOADED               0818.000
                                                        ACK.INTR                                                    0819.000
0052 6 0 0 1 4 0 0 3 02 0 800 0 0 0 0 0 0 0 0                   NOD=@00800000&FR(INTRTAB); TEST FOR RTOM INTERRUPT         0820.000
0053 1 4 0 0 0 1 C 0 00 F 7A5 0 0 0 0 0 0 0 0  55              NU=%T(ZE), IF IOREGSRDY *HOP EXECUTE.ACK;                  0821.000
0054 0 6 7 0 0 1 C 0 00 F 329 0 0 0 0 0 0 0 0  329             NU=%T(ZE), *LINK COUNT.DOWN; GO WAIT FOR READY             0822.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL  32/75  CPU                                                    INTERRUPT  SPB
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    EXECUTE.ACK                                                    0823.000

0055 C 4 0 0 4 0 E 0 00 0 90F 0 0 0 0 0 0 0 0  5F       FULLMAR=R(RDEV), IF NALUZ *GO TO RSTX.ACK;BR IF RTOM        0824.000

0056 0 6 7 0 1 0 0 0 00 0 301 0 0 0 0 0 0 0 0  301      *LINK CMD.ARSTX;                                           0825.000

0057 9 4 3 0 1 0 0 0 00 0 ABB 0 0 0 0 0 0 0 0  58       DECRN, IF %IORETRY:IOCHBUSY *HOP INTR.WAIT.READY;          0826.000

0058 0 6 7 0 1 0 0 0 00 0 30C 0 0 0 0 0 0 0 0  30C      *LINK TEST.RETRY;                                          0827.000

0059 A 4 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0  56       IF %NCTRZ *GO TO EXECUTE.ACK+1; RETRY 256 TIMES            0828.000
                                                *       TOTAL RETRY TIME = 2816 CLOCKS OR 422.4 MICRO  SECONDS     0830.000

                                                    INTR.ERR1                                                      0832.000

005A 0 6 0 0 1 0 0 0 00 0 520 0 0 0 0 0 0 0 0  520      *GO TO INT.IO.ERR1;                                        0833.000

                                                    INTR.WAIT.READY                                                0834.000

005B 1 4 0 0 0 1 C 0 00 F 5BA 0 0 0 0 0 0 0 0  5A       NU=%T(ZE), IF IONORESP:IOTIMEOUT *GO TO INTR.ERR1;         0835.000
                                                *                ; SET READY TIMEOUT COUNT FOR 512 CLOCKS (76.8US) 0837.000

005C 1 4 0 0 1 0 0 0 00 0 7AF 0 0 0 0 0 0 0 0  5F       IF IORESPRDY *HOP RSTX.ACK;  WAIT FOR READY                0839.000

005D A 4 3 0 1 0 0 0 00 0 90C 0 0 0 0 0 0 0 0  5C       DECRN, IF %NCTR7 *GO TO $-1;  WAIT FOR READY               0840.000

                                                    INTR.ERR2                                                      0841.000

005E 0 6 0 0 1 0 0 0 00 0 52C 0 0 0 0 0 0 0 0  52C      *GO TO INT.RDY.TIMEOUT; GO PROCESS I/O ERROR                0842.000

                                                    RSTX.ACK                                                       0843.000

005F 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0           NU=@0F000000;   SET RETRY COUNT TO 16                      0844.000

0060 0 6 7 0 1 0 1 0 00 0 305 0 0 0 0 0 0 0 0  305      S=N, *LINK CMD.RSTX; SAVE RETRY COUNT IN S REG             0845.000

0061 9 4 0 0 0 0 C 0 00 F 2B6 0 0 0 0 0 0 0 0  66       NU=T(ZE), IF %IONORESP:IOCHBUSY:IORETRY *HOP TEST.75.INT;  0846.000

0062 9 4 0 0 1 E C 0 00 0 EB5 0 0 0 0 0 0 0 0  65       NU=S, IF %IORETRY *HOP INTERRUPT.ERROR; RESTORE RETRY COUNT 0847.000

0063 0 6 7 0 1 0 0 0 00 0 30C 0 0 0 0 0 0 0 0  30C      *LINK TEST.RETRY; GO TEST RETRY                            0848.000

0064 A 4 3 0 1 0 0 0 00 0 900 0 0 0 0 0 0 0 0  60       DECRN, IF %NCTRZ *GO TO RSTX.ACK+1;  RETRY 16 TIMES        0849.000
                                                *       TOTALRETRY TIME = 192 CLOCKS OR 28.8 MICRO SECONDS         0851.000

                                                    INTERRUPT.ERROR                                                0853.000

0065 0 6 0 0 1 0 0 0 00 0 529 0 0 0 0 0 0 0 0  529      *GO TO INT.IO.ERR2;  GO PROCESS I/O ERROR                   0854.000

                                                    TEST.75.INT                                                    0855.000

0066 4 6 7 0 1 0 0 0 00 C C4F 0 0 0 0 0 0 0 0  C4F      IF MODE75S *LINK HANDLE.75.INT; EXIT FOR 75 INT HANDLER     0856.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          I N T E R R U P T  S P B
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                        SCHEDULE                                                                0857.000

0067 6 0 0 1 4 8 0 2 02 8 270 0 0 0 0 0 0 0 0                   NOD=@00270000!FR(INTR), PCTOMAR; TEST FOR CALM INTERRUPT                0858.000

0068 0 0 0 0 5 0 F 0 00 0 800 0 0 0 0 0 0 0 0                   T=R(INTRTAB,HWS); TEST FOR CALM RHFLAG                                  0859.000

0069 8 5 7 0 0 0 0 0 00 8 086 0 0 0 0 0 0 0 0   86              NOD=T, PCTOMAR, IF ALUZ *LINK CALM.TST; BRANCH IF CALM INT              0860.000

                                                        CLASSF.INT.RTN                                                          0861.000

                                                        SCHEDULE.55.TRAP                                                        0862.000

                                                        SCHEDULE.75.TRAP                                                        0863.000

006A 0 0 0 4 1 D F 0 00 D 800 0 0 0 0 0 0 0 0                   T=R(PCMASK)&MAR, OTHERBANK; GET CURRENT PC VALUE                        0864.000

006B 6 0 0 0 4 0 0 7 00 D 004 0 0 0 0 0 0 0 0                   NOD=FR(TRACE), OTHERBANK, RESET(HIREG); TEST FOR RHFLAG                 0865.000

006C 3 4 0 0 4 0 1 0 00 8 20F 0 0 0 0 0 0 0 0   6E              S=R(CPSTS), IF BMUX16 *GO TO $+2; GET CURRENT STATUS AND                0866.000

                                                        *                       ; BRANCH IF NOT RHFLAG IN TRACE                 0867.000

006D 0 0 0 3 0 A F 0 02 0 020 0 0 0 0 0 0 0 0                   T=@00000002:T; SET PC TO RIGHT HALFWORD                                 0868.000

006E 0 0 0 0 4 D 1 0 00 0 900 0 0 0 0 0 0 0 0                   S=S&R(STMASK); CLEAN UP CURRENT STATUS                                  0869.000

006F 0 0 0 0 4 0 E 0 00 D 600 0 0 0 0 0 0 0 0                   FULLMAR=R(SI.VECTOR), OTHERBANK; LOAD CONTEXT BLOCK ADDR                0870.000

0070 1 4 0 0 0 0 C 0 00 F 800 0 0 0 0 0 0 0 0   70              NU=T(ZE), IF BIBUSY *HOP $;WAIT FOR CHANNEL DRT IF STILL PENDING        0871.000

0071 9 4 0 0 0 A F 0 00 6 083 0 0 0 0 0 0 0 0   73              T=S:1, SDEST, IF %10TIMEOUT *HOP STORE.PSW1; FORMAT OLD PSW 1           0872.000

                                                        INTR.ERR4                                                               0873.000

0072 0 6 0 0 1 0 0 0 00 0 52D 0 0 0 0 0 0 0 0   52D             *GO TO INT.DRT.TIMEOUT;  GO PROCESS I/O ERROR                           0874.000

                                                                                                                                0875.000
                                                        **                                                                      0876.000
                                                        **      ENTRY PARAMETERS :                                              0877.000
                                                        **        MAR = SI VECTOR                                               0878.000
                                                        **        T = INTERRUPTED PSW WORD 1 (OLD PSW1)                         0879.000
                                                        **        S = 0  (NEW STATUS- CC'S & EXTENDED BIT)                      0880.000
                                                        **                                                                      0881.000

                                                                                                                                0882.000

                                                        STORE.PSW1                                                              0883.000

0073 4 6 7 0 0 8 1 0 00 A 201 0 0 0 0 0 0 0 0   201             S=S!T,IF ENBL.AEXP *LINK SET.PSD1.AFXP;CLEAR NEW STATUS FOR 55          0884.000

0074 0 0 0 0 1 0 0 0 1F 1 58C 0 0 0 0 0 0 0 0                   WRITE, FRCWORD, SETXCC(S); STORE INTR PSW AND SET NEW STATUS            0885.000

                                                                                                                                0886.000
                                                        **                                                                      0887.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTERRUPT SPB

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                      **          FIRST INSTRUCTION EXECUTED AFTER AN INTERRUPT/TRAP          0888.000
                                                      **          IS NONINTERRUPTABLE.  INTERRUPT VECTOR CONTENTS             0889.000
                                                      **          + 4 IS THE FIRST EXECUTABLE INSTRUCTION.                    0890.000
                                                      **          REFILL PIPELINE AND DISABLE ALL EXTERNAL EVENTS             0891.000
                                                      **          PRIOR TO THE *JUMP7.  CURRENT PROGRAM STATUS IS             0892.000
                                                      **          STORED AT THE LOCATION POINTED TO BY THE CONTENTS           0893.000
                                                      **          OF THE INTERRUPT VECTOR LOCATION.                           0894.000
                                                                                                                              0895.000
0075 0 0 0 3 1 3 2 0 02 0 040 0 0 0 0 0 0 0 0                         PC=@00000004+MAR;  INCREMENT PC                             0896.000
0076 4 5 7 0 5 0 F 0 00 C F94 0 0 0 0 0 0 0 0   94                    T=R(TRACE,HWS), IF MODE75S *LINK HANDLE.75.PSW;              0897.000
                                                      ***                                                                     0899.000
                                                      ***     RETURN PARAMETER :                                              0900.000
                                                      ***       T = R(TRACE,HWS)                                              0901.000
                                                      ***                                                                     0902.000
                                                      UNBLOCKED.75.RETURN                                                     0903.000
0077 6 0 0 1 4 C 1 1 02 D 7F0 0 0 0 0 0 0 0 0                         S=@007F0000&%FR(RDEV), OTHERBANK; GET ONES-COMP OF INT LVL   0905.000
0078 0 0 0 0 0 E 8 0 00 0 C00 0 0 0 0 0 0 0 0                         R(ALEVEL)=S, BMUX=T; SAVE NEW ACTIVE LEVEL & TEST R(TRACE,HWS) 0906.000
0079 5 5 0 0 1 0 0 0 00 B 086 0 0 0 0 0 0 0 0   86                    IF %BMUX19 *GO TO TRAP.INT.EXIT; EXIT IF TRAP BIT            0907.000
007A 6 0 0 0 4 D 0 3 02 D 020 0 0 0 0 0 0 0 0                         NOD=@02000000&FR(INTRTAB),OTHERBANK;   TEST RI PENDING BIT   0908.000
007B 6 0 0 4 4 D F 3 02 D FD0 0 0 0 0 0 0 0 0                         T=@FDFFFFFF&FR(INTRTAB),OTHERBANK;     CLEAR RI PENDING BIT  0909.000
007C C 4 0 0 0 A F 0 02 0 04F 0 0 0 0 0 0 0 0   7E                    T=@04000000:T,   IF NALUZ *GO TO RI.PENDING;  SET ACTIVE     0910.000
007D 0 0 0 4 0 D F 0 02 0 BF0 0 0 0 0 0 0 0 0                         T=@BFFFFFFF&T; CLEAR I/O IN PROGRESS BIT                     0911.000
                                                      RI.PENDING                                                              0912.000
007E 0 0 0 0 4 0 1 0 00 D D00 0 0 0 0 0 0 0 0                         S=R(INTRLOC), OTHERBANK; FETCH INTR ENTRY SCRATCH PAD ADDR    0913.000
007F 0 0 0 0 0 0 4 0 00 A 000 0 0 0 0 0 0 0 0                         SCRATCH(S)=T;  STORE INTERRUPT ENTRY                         0914.000
0080 0 0 0 1 1 0 1 0 02 A 850 0 0 0 0 0 0 0 0                         S=SCRATCH(@85);  FETCH ACTIVE INTERRUPT COUNT                0915.000
0081 0 0 0 0 1 6 F 0 16 0 D04 0 0 0 0 0 0 0 0                         T=S+1,RSTAEXP,SET(DINTRA);BUMP ACTIVE COUNT & SET DISP INT ACT 0916.000
0082 0 0 0 1 0 0 4 0 02 A 850 0 0 0 0 0 0 0 0                         SCRATCH(@85)=T;  STORE UPDATED ACTIVE INTERRUPT COUNT        0917.000
                                                      INTERRUPT.EXIT                                                          0918.000
0083 6 0 0 6 4 D B 7 02 0 000 0 0 0 0 0 0 0 0                         FR(TRACE)=@FFFF00FF&FR(TRACE); CLEAR RHFLAG, PWR FAIL, AND   0919.000
                                                      *                   BLOCK BIT AND 75 TRAP FLAG                          0920.000
0084 4 6 7 0 1 0 0 0 00 D 203 0 0 0 0 0 0 0 0   203                   IF IPU.HALT *LINK SIPU.RETURN;   RETURN TO INTERNAL EVENT    0921.000
0085 0 5 0 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0   F0                    *GO TO FILL.PIPELINE; EXIT IF BLOCK INTERRUPT MODE           0922.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    INTERRUPT SPB
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                        TRAP.INT.EXIT                                                    0923.000
0086 0 4 0 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0  83       *HOP INTERRUPT.EXIT;  DELAY REQUIRED                             0924.000
                                                        TRAP                                                             0925.000
0087 3 6 7 0 1 0 0 0 00 4 32E 0 0 0 0 0 0 0 0  32E      IF BMUX00 *LINK RAM.LOAD; BRANCH IF NOT RAM LOADED FLAG          0926.000
0088 6 0 0 0 4 A B 3 02 0 040 0 0 0 0 0 0 0 0           FR(INTRTAB)=@04000000:FR(INTRTAB); SET ACTIVE FLAG               0927.000
0089 0 0 0 3 1 E F 0 02 0 040 0 0 0 0 0 0 0 0           T=@00000004; SET ACTIVATE INTERRUPT CODE                         0928.000
008A 0 0 0 0 4 0 E 0 00 0 900 0 0 0 0 0 0 0 0           FULLMAR=R(RDEV);                                                 0929.000
008B 0 6 7 0 1 0 0 0 00 0 375 0 0 0 0 0 0 0 0  375      *LINK ISSUE.ICT;                                                 0930.000
008C A 5 0 0 1 0 0 0 00 8 965 0 0 0 0 0 0 0 0  65       PCTOMAR, IF %NCTR7 *GO TO INTERRUPT.ERROR; EXIT IF RTOM ERROR    0931.000
008D 6 0 0 0 4 0 0 7 00 0 000 0 0 0 0 0 0 0 0           NOD=FR(TRACE), OTHERBANK; TEST POWER FAIL/SYS OVRD FLAG          0932.000
008E 3 5 0 0 1 0 0 0 00 8 06A 0 0 0 0 0 0 0 0  6A       IF BMUX19 *GO TO SCHEDULE.55.TRAP; EXIT IF NO PF/SYS OVRD        0933.000
008F 6 0 0 3 4 3 E 3 02 F 040 0 0 0 0 0 0 0 0           FULLMAR=@00000004+FR(INTRTAB,7E); COMPUTE TRAP SI VECTOR ADDR    0934.000
0090 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B      *LINK MEMORY.READ; GO FETCH TRAP SI VECTOR (PF OR SYS OVRD ONLY) 0935.000
0091 0 0 0 4 3 0 F 0 00 0 800 0 0 0 0 0 0 0 0           T=R(PCMASK)&0I, OTHERBANK; GET SI VECTOR                         0936.000
0092 0 0 0 0 0 0 8 0 00 8 600 0 0 0 0 0 0 0 0           R(ST.VECTOR)=T, PCTOMAR;SAVE PWR FAIL OR SYS OVRIDE TRAP VECT    0937.000
0093 0 5 0 0 1 0 0 0 00 0 06A 0 0 0 0 0 0 0 0  6A       *GO TO SCHEDULE.55.TRAP;                                         0938.000
                                                                                                                         0939.000
                                                        HANDLE.75.PSW                                                    0940.000
0094 0 0 0 1 1 0 1 0 02 A 900 0 0 0 0 0 0 0 0           S=SCRATCH(@90);  FETCH CURRENT PSW WORD 2                        0941.000
0095 0 0 0 3 1 3 E 0 02 0 080 0 0 0 0 0 0 0 0           FULLMAR=@00000008+MAR; VECTOR TO NEW PSW WORD 1                  0942.000
0096 0 0 0 0 1 E F 0 1C 1 080 0 0 0 0 0 0 0 0           T=S, READ, FRCWORD; FETCH NEW PSW WORD 1                         0943.000
0097 9 4 0 0 1 0 0 0 00 0 559 0 0 0 0 0 0 0 0  99       IF %OPNORESP:OPRNOPE:OPTIMEOUT *HOP $+2;                         0944.000
                                                        INTR.ERR3                                                        0945.000
0098 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(7E), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR         0946.000
0099 0 0 0 0 4 0 0 0 00 0 F00 0 0 0 0 0 0 0 0           NOD=R(TRACE); TEST FOR BLOCKED INTERRUPT FLAG                    0947.000
009A 3 4 0 6 0 D F 0 02 9 3FC 0 0 0 0 0 0 0 0  9C       T=@FFFF3FFF&T, IF BMUX17 *HOP $+2; CLEAR INT FLAGS 0.PSW2        0948.000
009B 0 0 0 2 0 A F 0 02 0 400 0 0 0 0 0 0 0 0           T=@00004000:T; SET BLOCKED INT MODE FLAG                         0949.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    INTERRUPT SPR
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

009C 0 0 0 7 1 3 E 0 00 0 FC0 0 0 0 0 0 0 0 0          FULLMAR=@FFFFFFFC+MAR; VECTOR TO OLD PSW 2                    0950.000
009D 0 0 0 0 3 0 1 0 1E 1 386 0 0 0 0 0 0 0 0          S=DI, WRITE, FRCWORD, RESET(ENBL.AEXP); STORE OLD PSW 2       0951.000
009E 0 0 0 0 3 D 0 0 02 0 010 0 0 0 0 0 0 0 0          NOD=@01000000&DI; TEST FOR ENABLE AFXP IN N.PSW1              0952.000
009F 0 0 0 4 3 D 2 0 00 0 800 0 0 0 0 0 0 0 0          PC=R(PCMASK)&DI; LOAD PC FROM N.PSW1                          0953.000
00A0 C 0 0 0 1 0 0 0 00 0 806 0 0 0 0 0 0 0 0          IF NALUZ SET(ENBL.AFXP);                                      0954.000
00A1 6 0 0 2 4 D 0 7 02 0 040 0 0 0 0 0 0 0 0          NOD=@00000400&FR(TRACE); TEST FOR 75 TRAP FLAG                0955.000
00A2 0 0 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0          SETXCC(S);  SET NEW CC'S, EXT BIT FROM N.PSW1                 0956.000
00A3 8 4 0 0 4 0 6 0 00 D 00B 0 0 0 0 0 0 0 0  AB      DI=R(N.PSW2), OTHERBANK, IF ALUZ *GO TO TEST.MAPPED.INTERRUPT; 0957.000
                                                  STORE.CPU.STATUS.OR.NUM                                       0958.000
00A4 0 0 0 3 1 E 1 0 02 0 100 0 0 0 0 0 0 0 0          S=@00000010; SET STATUS ADDR OR SVC CALL # BIAS               0959.000
00A5 0 0 0 0 4 3 E 0 00 D 600 0 0 0 0 0 0 0 0          FULLMAR=S+R(ST.VECTOR), OTHERBANK; BIAS TO STATUS ADDRESS     0960.000
00A6 0 0 0 0 4 0 F 0 00 D 800 0 0 0 0 0 0 0 0          T=R(TBL2), OTHERBANK; FETCH CPU STATUS WORD OR SVC CALL #.    0961.000
                                                  STORE.STATUS                                                  0962.000
00A7 0 0 0 0 3 0 0 0 1E 1 080 0 0 0 0 0 0 0 0          NOD=DI, WRITE, FRCWORD;   STORE SVC CALL # OR CPU STATUS WORD 0963.000
00A8 5 4 0 0 3 0 0 0 00 8 008 0 0 0 0 0 0 0 0  Ab      NOD=DI, IF %BMUX16 *GO TO STORE.STATUS.EXIT;                  0964.000
00A9 0 0 0 0 3 0 0 0 00 0 906 0 0 0 0 0 0 0 0          NOD=DI, SET(UNBLOCK);  REFRESH BLOCK MODE TIMEOUT COUNT       0965.000
00AA 5 0 0 0 3 0 0 0 00 9 106 0 0 0 0 0 0 0 0          NOD=DI, IF %BMUX17 RESET(UNBLOCK);                            0966.000
                                                  STORE.STATUS.EXIT                                             0967.000
                                                  TEST.MAPPED.INTERRUPT                                         0968.000
00AB 0 6 7 0 3 0 F 0 00 0 9D3 0 0 0 0 0 0 0 0  9D3     T=DI, *LINK EVALUATE.PSD2.MAP; GET N.PSW2 IN T                0969.000
00AC 4 5 0 0 5 0 F 0 00 8 F77 0 0 0 0 0 0 0 0  77      T=R(TRACE,HWS), IF UNBLOCK *GO TO UNBLOCKED.75.RETURN;        0970.000
00AD 0 5 0 0 1 0 0 0 00 0 083 0 0 0 0 0 0 0 0  83      *GO TO INTERRUPT.EXIT;                                        0971.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTERRUPT SPB
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                    0972.000
                                                                                                                                    0973.000
                                                                ** IPU TRAP                                                         0974.000
                                                                **  THIS TRAP OCCURS WHEN THE IPU CAUSES A TRAP TO THE CPU          0975.000
                                                                *                                                                   0976.000
                                                                IPU.TRAP                                                            0977.000
                                                                *                                                                   0978.000
00AE 4 5 0 0 1 0 0 0 00 2 080 0 0 0 0 0 0 0 0   80      IF IPU *GOTO $+2;                                                           0979.000
00AF 0 6 0 0 1 0 0 0 13 0 580 0 0 0 0 0 0 0 0   580     RSTIPUFLG,*GOTO TRAP.MODE75;    TRAP CPU                                    0980.000
00B0 0 0 0 0 1 0 0 0 00 0 806 0 0 0 0 0 0 0 0           SET(MODE75);                                                                0981.000
00B1 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0           SET(ENAUORD);                                                               0982.000
00B2 0 0 0 0 1 0 0 0 00 1 005 0 0 0 0 0 0 0 0           RUN,RESET(ENAUORD);             HALT/RUN TRANSITION                         0983.000
00B3 0 0 0 3 0 A F 0 02 6 F40 0 0 0 0 0 0 0 0           T=@000000E4:1,SDEST;            IPU START TRAP                              0984.000
00B4 0 0 0 0 1 0 0 0 00 0 304 0 0 0 0 0 0 0 0           RESET(TRACEFF);                                                             0985.000
00B5 0 6 0 0 1 0 0 0 13 0 580 0 0 0 0 0 0 0 0   580     RSTIPUFLG,*GOTO TRAP.MODE75;                                                0986.000
```

```
          S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                          INTERRUPT  SPB
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                        0987.000

                                                    (@B6)                                               0988.000

                                                                                                        0989.000
                                                    **                                                  0990.000
                                                    **  THIS CHECKS CALM INTERRUPT FLAGS FOR A CALM RHFLAG   0991.000
                                                    **    IF THE FLAG IS FOUND, PSW BIT 06 IS SET       0992.000
                                                    **                                                  0993.000

                                                                                                        0994.000

                                                    CALM.TST                                            0995.000

00B6 3 1 0 0 1 E F 0 02 A 020 0 0 0 0 0 0 0 0           IF BMUX18 *JUMPJ, T=@02000000; RETURN IF NOT CALM RHFLAG   0996.000

00B7 0 4 7 4 0 A 8 0 00 0 209 0 0 0 0 0 0 0 0   B9      R(CPSTS)=R(CPSTS):T, *LINK DUD.CALM; SET PSW BIT 06        0997.000

00B8 6 0 0 4 4 D B 3 02 0 0F0 0 0 0 0 0 0 0 0           FR(INTRTAB)=@0FFFFFFF&FR(INTRTAB); CLR CALM RHFLAG  0998.000

                                                    DUD.CALM                                            0999.000

00B9 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPJ;                                             1000.000

                                                    NPM.CHECK                                           1001.000

00BA 0 0 0 0 0 D 0 0 02 0 010 0 0 0 0 0 0 0 0           NOD=@01000000&T; TEST FOR SAVED NPM FLAG            1002.000

00BB 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0           RESET(ENAUORD);                                     1003.000

00BC B 5 0 0 5 0 F 0 00 0 F10 0 0 0 0 0 0 0 0   10      T=R(TRACE,HWS), IF ALUZ *GO TO MISC.INTERRUPTS; EXIT IF NO NPM   1004.000

00BD 6 0 0 0 4 B B 7 02 0 010 0 0 0 0 0 0 0 0           FR(TRACE)=%@01000000&FR(TRACE); CLEAR SAVED NPM BIT  1005.000

00BE 0 5 0 0 1 0 0 0 00 0 034 0 0 0 0 0 0 0 0   34      *GO TO EXTG.NPM.ENTRY; GO PROCESS NPM               1006.000
```

```
          S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    P A N E L   S C H E D U L I N G
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1007.000
                                                        STOP                                                        1008.000
00BF 9 4 0 0 0 0 0 0 00 0 EA1 0 0 0 0 0 0 0 0   C1          NOD=T, IF %IPLSW *GO TO $+2;                            1009.000
00C0 0 6 0 0 1 0 0 0 00 0 15E 0 0 0 0 0 0 0 0  15E          *GO TO IPL;                                             1010.000
00C1 1 4 0 0 0 0 0 0 00 0 F29 0 0 0 0 0 0 0 0   C9          NOD=T, IF FFSYSR *GO TO PANEL.SYS.RES;                  1011.000
00C2 9 4 0 0 0 0 0 0 00 0 F8E 0 0 0 0 0 0 0 0   CE          NOD=T, IF %TRACF *GO TO PANEL.HALT;                     1012.000
00C3 9 4 0 0 0 0 0 0 00 0 F46 0 0 0 0 0 0 0 0   C6          NOD=T, IF %SERIAL.PANEL *HOP $+3;                       1013.000
00C4 0 6 7 0 1 0 0 0 00 0 A5F 0 0 0 0 0 0 0 0  A5F          *LINK S.P.ATTN;                                         1014.000
00C5 0 4 0 0 4 0 F 0 00 0 F07 0 0 0 0 0 0 0 0   C7          T=R(TRACE), *GO TO $+2;                                 1015.000
00C6 1 4 0 0 0 0 0 0 00 0 F9F 0 0 0 0 0 0 0 0   CE          NOD=T, IF PPATTN *HOP PANEL.HALT;                       1016.000
00C7 3 4 0 0 1 0 0 0 00 4 00E 0 0 0 0 0 0 0 0   CE          IF BMUX00 *HOP PANEL.HALT;                              1017.000
00C8 0 6 0 0 1 0 0 0 00 0 F6F 0 0 0 0 0 0 0 0  F6F          *GOTO EXTERNAL.TPU.TEST;    CPU ALREADY HALTED          1018.000
                                                        PANEL.SYS.RES                                               1019.000
00C9 0 6 7 0 1 0 0 0 14 0 120 0 0 0 0 0 0 0 0  120          UPACK, *LINK SYSTEM.RESET;                              1020.000
00CA 0 6 7 0 1 0 0 0 00 0 26F 0 0 0 0 0 0 0 0  26F          *LINK PP.CSW.ADDR;                                      1021.000
00CB 0 6 7 0 1 0 0 0 00 0 267 0 0 0 0 0 0 0 0  267          *LINK PP.CSW.ADDR1;                                     1022.000
00CC 9 4 0 0 1 0 0 0 00 0 F4E 0 0 0 0 0 0 0 0   CE          IF %SERIAL.PANEL *HOP $+2;                              1023.000
00CD 0 6 7 0 0 1 C 0 00 F AF3 0 0 0 0 0 0 0 0  AF3          NU=%T(ZE), *LINK S.P.CLEAR;                             1024.000
                                                        PANEL.HALT                                                  1025.000
00CE 0 0 0 3 1 E 1 0 02 0 110 0 0 0 0 0 0 0 0               S=@00000011; SET PANEL STOP CODE IN S.                  1026.000
00CF 6 0 0 4 4 D F 7 02 0 FF0 0 0 0 0 0 0 0 0               T=@FFFFFFFF&FR(TRACE);          CLEAR STEP BIT          1027.000
00D0 6 0 0 0 0 A B 7 02 0 800 0 0 0 0 0 0 0 0               FR(TRACE)=@80000000:T;          SET HALT BIT            1028.000
00D1 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0               SET(ENAUORD);                                           1029.000
00D2 0 0 0 0 1 0 0 0 00 2 804 0 0 0 0 0 0 0 0               HALT,SET(TRACEFF);                                      1030.000
00D3 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0               RESET(ENAUORD);                                         1031.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
PANEL SCHEDULING
SFL 32/75 CPU
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                    1032.000
                                                                    1033.000
PANEL.ATTN
```

```
                                                                    CHECK.STOP                                                                  1034.000
00D4 0 0 0 0 1 0 0 0 14 0 804 0 0 0 0 0 0 0 0                               UPACK , SET(HIREG); ACKNOWLEDGE PANEL                               1035.000
00D5 1 4 0 0 1 E 3 0 00 0 EF7 0 0 0 0 0 0 0 0   D7                          MAR=S, IF ADDRSTOP *HOP $+2; DUMMY ADR TO MAR                       1036.000
00D6 0 6 0 3 1 E 1 0 00 0 208 0 0 0 0 0 0 0 0   208                         S=SNIBR, *GO TO PPATTN.HALT2; MAKE SEND FUNCTION CODE IN S          1037.000
00D7 0 6 0 0 1 0 0 0 00 0 20A 0 0 0 0 0 0 0 0   20A                         *GO TO PPATTN.HALT1;                                                1038.000
                                                                    ADDR.STOP                                                                   1039.000
00D8 0 6 7 0 1 0 0 0 00 0 117 0 0 0 0 0 0 0 0   117                         *LINK BACKDATE.PC;                                                  1040.000
00D9 0 5 0 0 1 0 0 0 00 0 0CE 0 0 0 0 0 0 0 0   CE                          *GO TO PANEL.HALT;                                                  1041.000
                                                                    **              TRANSITION FROM RUN TO HALT                                 1042.000
                                                                    TRACE.RUN                                                                   1043.000
00DA 5 5 0 0 0 0 1 0 00 4 0E1 0 0 0 0 0 0 0 0   E1                          S=T,IF %BMUX00 *GO TO RUN.HALT;                                     1044.000
00DB 6 0 0 3 4 0 0 7 02 0 080 0 0 0 0 0 0 0 0                               NOD=@00000008&FR(TRACE);        ENTRY FROM EXTG                     1045.000
00DC 0 0 0 1 1 E 0 0 00 0 000 0 0 0 0 0 0 0 0                               NOD=SLEFT;      TEST STEP FLAG                                      1046.000
00DD B 5 0 1 1 E 0 0 00 0 0E6 0 0 0 0 0 0 0 0   E6                          NOD=SLEFT,IF ALUZ *GO TO TRACE.INCOMPLETE; NOT ENTRY FROM EXTG      1047.000
00DE A 5 0 0 1 0 0 0 00 0 8E2 0 0 0 0 0 0 0 0   E2                          IF %ALUNEG *GO TO RUN.HALT+1; NOT STEP BIT                          1048.000
                                                                    TRACE.STEP                                                                  1049.000
00DF 6 0 0 0 4 B B 7 02 0 400 0 0 0 0 0 0 0 0                               FR(TRACE)=%@40000000&FR(TRACE); CLEAR STEP BIT                      1050.000
                                                                    **              ENTRY FROM INSTRUCTION STEP EXECUTION                       1051.000
00E0 0 6 0 0 1 0 0 0 00 0 C37 0 0 0 0 0 0 0 0   C37                         *GO TO STEP.HALT;                                                   1052.000
                                                                    **                      RUN AND HALT BIT SET                                1053.000
                                                                    RUN.HALT                                                                    1054.000
                                                                    **              TRANSITION FROM HALT TO RUN                                 1055.000
00E1 0 0 0 4 0 0 F 0 02 0 7F0 0 0 0 0 0 0 0 0                               T=@7FFFFFFF&T;  CLEAR HALT BIT IN R(TRACE)                          1056.000
00E2 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0                               SET(ENAUORD);                                                       1057.000
00E3 0 0 0 0 1 0 0 0 00 1 304 0 0 0 0 0 0 0 0                               RUN,RESET(TRACEFF);                                                 1058.000
00E4 0 5 7 0 0 0 8 0 14 0 F0F 0 0 0 0 0 0 0 0   F                           UPACK, R(TRACE)=T, *LINK DUD.0XX;                                   1059.000
00E5 0 5 0 0 1 0 0 0 00 0 0BA 0 0 0 0 0 0 0 0   BA                          *GO TO NPM.CHECK; GO CHECK FOR SAVED ERRORS                         1060.000
                                                                    TRACE.INCOMPLETE                                                            1061.000
```

```
00E6 6 0 0 2 4 B B 7 02 0 600 0 0 0 0 0 0 0 0                FR(TRACE)=%@00006000&FR(TRACE);                                  1062.000

00F7 5 0 0 0 1 0 0 0 00 2 304 0 0 0 0 0 0 0 0                IF %ALUNEGW RESET(TRACEFF); TEST INST. STEP BIT IN R(TRACE)      1063.000

00F8 0 5 0 0 5 0 F 0 00 0 F10 0 0 0 0 0 0 0 0   10           T=R(TRACE,HWS), *GO TO MISC.INTERRUPTS;                          1064.000

                                                                                                                              1065.000
                                                     **            ENTRY FROM OTHERS                                          1066.000
                                                     **                  UNDEFINED INSTRUCTION   (PRIOR INSTR)              1067.000
                                                     **                  PRIVELIGE VIOLATION     (PRIOR INSTR)              1068.000
                                                     **                                                                     1069.000
                                                                                                                              1070.000
                                                     OTHERS.TEST                                                              1071.000

00E9 0 4 7 4 4 B 0 0 02 0 FBE 0 0 0 0 0 0 0 0   FE           NOD=%@FBFFFFFF&R(TRACE), *LINK DUD.0EX; TEST UNDEF INSTR         1072.000

00EA C 6 7 0 1 0 0 0 00 0 560 0 0 0 0 0 0 0 0   560          IF NALUZ *LINK UNDEFINED.INSTRUCTION;                            1073.000

00EB 0 4 7 4 4 B 0 0 02 0 F7E 0 0 0 0 0 0 0 0   FE           NOD=%@F7FFFFFF&R(TRACE), *LINK DUD.0EX; TEST PRIV VIOLATION      1074.000

00EC C 6 7 0 1 0 0 0 00 0 551 0 0 0 0 0 0 0 0   551          IF NALUZ *LINK PRIVILEGE.VIOLATION;                              1075.000

00ED 0 5 0 0 1 0 0 0 00 0 03F 0 0 0 0 0 0 0 0   3F           *GO TO TEST.FFINT;                                               1076.000

                                                     DUD.0EX                                                                  1077.000

00EE 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                *JUMPJ;                                                          1078.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FILL.PIPFLINE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    1079.000
                                                        RENABLE.EXIT                                                1080.000
00EF 0 0 0 0 1 0 0 0 00 0 F04 0 0 0 0 0 0 0 0           SET(ENAINTFF);  ENABLE INTERRUPTS                           1081.000
                                                        FILL.PIPFLINE                                               1082.000
00F0 0 0 0 3 1 E C 0 02 7 004 0 0 0 0 0 0 0 0           I1T010,NOD=0,RESET(HIREG);              FLUSH PIPFLINE      1083.000
00F1 1 0 0 0 1 E E 0 02 0 EFD 0 0 0 0 0 0 0 0           FULLMAR=@EF000000,IF ADDRSTOP CLDNU;    ADDRESS STOP REDISPATCH 1084.000
00F2 1 0 0 0 1 E 0 0 02 0 DFD 0 0 0 0 0 0 0 0           NOD=@DF000000,IF INTRENA CLDNU;         INTERRUPT DISPATCH  1085.000
00F3 6 0 0 0 4 B F 7 02 0 130 0 0 0 0 0 0 0 0           T=%@13000000&FR(TRACE);                 CLEAR PRIOR ERROR STAT 1086.000
00F4 A 0 0 0 0 0 C 0 19 F 983 0 0 0 0 0 0 0 0           NU=T(ZE),FETCHPC,IF %NCTRZ IGNSTOP;                         1087.000
00F5 0 0 0 3 0 B F 0 02 0 FF0 0 0 0 0 0 0 0 0           T=%@000000FF&T;                 RESET ENTRY FLAGS           1088.000
00F6 5 4 0 2 0 B B 3 02 8 90A 0 0 0 0 0 0 0 0  FA       R(NCTR)=%@00009000&T,IF %RMUX16 *GO TO RTN.RHALF;           1089.000
00F7 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0           FETCHPC;                                                    1090.000
00F8 F 3 4 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0           DECODE(0),I1T010,IF FALSE *JUMP7;                           1091.000
00F9 0 2 6 0 1 0 F 0 00 0 104 0 0 0 0 0 0 0 0           T=MAR,DECODE(#),RESET(EXFF),*JUMPD;                         1092.000
                                                        RTN.RHALF                                                   1093.000
00FA 0 0 0 0 1 0 0 0 19 C 080 0 0 0 0 0 0 0 0           FETCHPC,SETRHF;                                             1094.000
00FB F 3 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0           I1T010,IF FALSE *JUMP7;                                     1095.000
00FC 0 0 4 0 2 0 0 0 00 9 040 0 0 0 0 0 0 0 0           DECODE(4),NOD=I0,SHIFTI0;                                   1096.000
00FD 0 2 6 0 2 0 0 0 00 0 104 0 0 0 0 0 0 0 0           *JUMPD,NOD=I0,DECODE(#),RESET(EXFF);                        1097.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                                    POWER  FAIL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1098.000
                                                        POWER.FAIL                                                  1099.000
00FE 0 0 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0               RESET(HIREG);                                           1100.000
00FF 0 0 0 0 0 0 C 0 00 F 805 0 0 0 0 0 0 0 0               NU=T(7E), SET(ENAUORD);                                 1101.000
0100 0 0 0 2 1 E 2 0 02 0 037 0 0 0 0 0 0 0 0               PC=@00000300, CLRS;                                     1102.000
0101 6 0 0 7 1 9 8 3 00 3 F80 0 0 0 0 0 0 0 0               FR(TEMP1)=%@FFFFFFFB,RESETIO; CLEAR I/O & CP FLAGS      1103.000
0102 6 0 0 1 1 E B 5 02 0 015 0 0 0 0 0 0 0 0               FR(TEMP3)=@00010000, RESET(ENAUORD); SET SCRATCH ADDR BUMP VALUE 1104.000
0103 1 4 0 0 1 E 6 0 00 8 F2B 0 0 0 0 0 0 0 0  10B          UI=S, PCTOMAR, IF FFSYSR *GO TO POWER.ON;               1105.000
                                                        POWER.OFF                                                   1106.000
0104 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0               S=SCRATCH(S);  FETCH SCRATCH PAD ENTRY                  1107.000
                                                        ROLL.OUT                                                    1108.000
0105 4 6 0 0 1 0 0 0 00 2 CDC 0 0 0 0 0 0 0 0  CDC          IF IPU *GOTO IPU.POWER.DWN ;       DON'T DUMP IPU SCRATCH PAD 1109.000
0106 0 0 0 0 1 E F 0 1F 1 08F 0 0 0 0 0 0 0 0               T=S, WRITE, FRCWORD, CDECRN;                            1110.000
0107 6 0 0 4 3 3 6 5 00 6 000 0 0 0 0 0 0 0 0               UI=FR(TEMP3)+DI, SDEST;  BUMP SCRATCH PAD ADDR          1111.000
0108 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0               S=SCRATCH(S);  FETCH NEXT SCRATCH PAD ENTRY              1112.000
0109 E 4 0 4 1 3 E 3 00 0 905 0 0 0 0 0 0 0 0  105          FULLMAR=FR(TEMP1)+MAR, IF %NCTRZ *GO TO ROLL.OUT; BUMP MAR+4 1113.000
010A 0 5 0 0 4 0 2 0 00 0 E10 0 0 0 0 0 0 0 0  110          PC=R(TEMP4), *GO TO PWR.OFF.OUT; RESTORE PC VALUE       1114.000
                                                        POWER.ON                                                    1115.000
010B 0 6 0 0 1 0 0 0 00 0 CD7 0 0 0 0 0 0 0 0  CD7            *GO TO IPU.POWER.ON; CHECK FOR IPU ON POWER UP        1116.000
                                                        POWER.READ.OK                                               1117.000
010C 0 0 0 3 1 3 2 0 02 0 04F 0 0 0 0 0 0 0 0               PC=@00000004+MAR,CDECRN;                                1118.000
010D 0 0 0 0 3 0 4 0 00 A 000 0 0 0 0 0 0 0 0               SCRATCH(S)=DI;                                          1119.000
010E A 4 0 0 0 3 1 0 00 8 90B 0 0 0 0 0 0 0 0  10B          S=S+T, PCTOMAR, IF %NCTRZ *GO TO POWER.ON;              1120.000
010F 0 6 0 0 0 0 F 0 00 F F3F 0 0 0 0 0 0 0 0  F3E          T=T(ZE), *GO TO POWER.UP.S.P.CLEAR; CLEAR POWER UP ERROR FLAGS 1121.000
                                                        PWR.OFF.OUT                                                 1122.000
0110 0 6 0 0 1 0 0 0 00 0 334 0 0 0 0 0 0 0 0  334            *GO TO IPU.PWR.OUT; CHECK FOR IPU ON POWER OUT        1123.000
0111 6 0 0 3 1 E B 2 02 6 003 0 0 0 0 0 0 0 0               FR(INTR)=@00000000,CLRSYSR, SDEST;     SET INTRRUPT LEVEL 0 1124.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
                                                                    P O W E R  F A I L
S E L  3 2 / 7 5  C P U
PC  T S M A B + 0 R  Y X PCH + B D M + B A D  ADDR

0112 6 0 0 4 3 D B 7 02 D 7F0 0 0 0 0 0 0 0 0        FR(TRACE)=@7FFFFFFF&DI,OTHERBANK;  CLEAR HALT BIT          1125.000
0113 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0        SET(ENAUORD);                                               1126.000
0114 0 0 0 0 1 0 0 0 00 1 005 0 0 0 0 0 0 0 0        RUN, RESET(ENAUORD);                                        1127.000
0115 0 0 0 0 0 1 0 0 00 F 906 0 0 0 0 0 0 0 0        NU=%T(ZE), SET(UNBLOCK);                                    1128.000
0116 0 6 0 0 1 0 0 0 00 0 572 0 0 0 0 0 0 0 0  572   *GO TO PF.QUEUE;                                            1129.000
                                                                                                              1130.000
                                                  BACKDATE.PC                                                 1131.000
0117 0 0 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0        RESET(HIREG);                                               1132.000
0118 0 0 0 0 1 0 0 0 00 0 704 0 0 0 0 0 0 0 0        RESET(ENAINTFF);                                            1133.000
0119 1 4 0 0 1 0 0 0 00 0 BD9 0 0 0 0 0 0 0 0  119   IF RIBUSY *GO TO $;                                         1134.000
011A A 4 0 0 1 0 0 0 0C 8 A0D 0 0 0 0 0 0 0 0  110   I1=S,RSTRHF,PCTOMAR,IF %RHFLAG *GO TO $+3;                  1135.000
011B 1 4 0 0 1 0 0 0 00 0 08D 0 0 0 0 0 0 0 0  110   IF EXFLAG *GO TO $+2;                                       1136.000
011C 6 0 0 2 4 A B 7 02 0 800 0 0 0 0 0 0 0 0        FR(TRACE)=@00006000:FR(TRACE);  REMEMBER RIGHT HAND FLAG    1137.000
011D 0 1 0 7 1 3 2 0 00 0 F00 0 0 0 0 0 0 0 0        *JUMPJ,PC=@FFFFFFF8+MAR;                BACKDATE PC        1138.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SYSTEM RESET

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1139.000

                                                                                                                    1140.000
                                        **                                                                          1141.000
                                        **          CLEAR GENERAL PURPOSE REGISTERS 0-7                             1142.000
                                        **                                                                          1143.000
                                        **    NOTE: THE REGISTERS CLEARED ARE ADDRESSED BY THE ONES-                1144.000
                                        **          COMPLEMENT OF N REG BITS 4-7.                                   1145.000
                                        **                                                                          1146.000

                                                                                                                    1147.000

                                        (@120)                                                                      1148.000

                                        SYSTEM.RESET                                                                1149.000

0120 0 0 0 0 1 F C 0 02 0 0F4 0 0 0 0 0 0 0 0       NH=@0F000000, RESET(HIREG); INITIALIZE REG 0 ADDR IN N BITS 4-7 1150.000

0121 0 0 0 7 1 9 E 0 11 0 FF7 0 0 0 0 0 0 0 0       FULLMAR=%@FFFFFFFF, RESETFF, CLRS; RESET ALL FLIP/FLOPS AND     1151.000

                                        *                 ; CLEAR MAR AND S REG.                                    1152.000

                                        LOOP.CLEAR.GPRS                                                             1153.000

0122 0 0 3 0 1 E B 3 00 0 000 0 0 0 0 0 0 0 0       R(NCTR)=S, DECRN;  CLEAR GPR ADDRESSED BY ONES-COMP OF N REG 4-7 1154.000

                                        *                       ;  AND INCREMENT REG ADDR  (DECREMENT N REG)      1155.000

0123 2 4 0 0 1 0 0 0 00 0 B02 0 0 0 0 0 0 0 0  122  NOD=N, IF NCTR4 *GO TO LOOP.CLEAR.GPRS; LOOP UNTIL R7 IS CLEARED 1156.000

                                                                                                                    1157.000
                                        **                                                                          1158.000
                                        **          SET SYSTEM CONSTANTS & CLEAR SCRATCH PAD FLAGS                  1159.000
                                        **                                                                          1160.000
                                        **          THIS ENTRY POINT BYPASSES THE CLEAR OF THE GPRS                 1161.000
                                        **                                                                          1162.000

                                                                                                                    1163.000

                                        SET.SYSTEM.CONSTANTS                                                        1164.000

0124 0 0 0 7 1 9 F 0 11 0 030 0 0 0 0 0 0 0 0       T=%@FFFFFF03, RESETFF; LOAD T = @000000FC & CLEAR ALL F/F'S     1165.000

0125 0 0 0 0 1 0 0 0 00 0 C05 0 0 0 0 0 0 0 0       SET(ENATBMT);  ENABLE UART TRANSMIT BUFFER EMPTY                 1166.000

0126 0 0 0 1 0 A F 0 02 0 070 0 0 0 0 0 0 0 0       T=@00070000:T;                                                  1167.000

0127 6 0 0 2 0 A B 0 02 0 FF0 0 0 0 0 0 0 0 0       FR(PCMASK)=@0000FF00:T; SET R(PCMASK) = @0007FFFC               1168.000

0128 6 0 0 0 1 E B 1 02 0 FE0 0 0 0 0 0 0 0 0       FR(STMASK)=@FE000000; SET R(STMASK) = @FE000000                 1169.000

0129 0 0 0 0 1 0 0 0 00 0 F06 0 0 0 0 0 0 0 0       SET(DIS.BLK.TIMEOUT); DISABLE BLOCK MODE TIMEOUT TRAP           1170.000

012A 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0       SET(ENAUORD);                                                   1171.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    S Y S T E M   R E S E T
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

012B 0 0 0 0 1 0 0 0 16 4 003 0 0 0 0 0 0 0 0          RSTPROTV,CLRSYSR, RSTAEXP; CLEAR MISC. FLAGS                    1172.000

                                                                                                                    1173.000
                                                ;;                                                                  1174.000
                                                ;;          CLEAR MEMORY PROTECT FLAGS                              1175.000
                                                ;;                                                                  1176.000
                                                ;;      CLEAR THE 16 PROTECT REGISTERS ADDRESSED BY MAR 05-08       1177.000
                                                ;;       WITH THE DATA (=0) FROM FILE OUTPUT BITS 16-31             1178.000
                                                ;;                                                                  1179.000

                                                                                                                    1180.000
012C 0 0 0 3 1 E E 0 02 0 005 0 0 0 0 0 0 0 0          FULLMAR=@00000000, RESET(ENAUORD); SET INITIAL PROT REG ADDR =0 1181.000

012D 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0          NU=@0F000000; SET INITIAL PROTECT REG COUNT                     1182.000

                                                PROT.MAP.CLEAR                                                      1183.000

012E 0 0 0 0 4 0 0 0 07 0 000 0 0 0 0 0 0 0 0          NOD=R(0), WRPMAP; CLEAR THE PROT REG ADDRESSED BY MAR 05-08     1184.000

012F 0 0 0 2 1 3 E 0 02 0 800 0 0 0 0 0 0 0 0          FULLMAR=@00008000+MAR; BUMP PROTECT REG ADDR IN MAR             1185.000

0130 A 5 3 0 1 0 0 0 00 0 92F 0 0 0 0 0 0 0 0  12E     NOD=N, DECRD, IF %NCTRZ *GO TO PROT.MAP.CLEAR; DECREMENT COUNT  1186.000

                                                                                                                    1187.000
                                                ;;                                                                  1188.000
                                                ;;          CLEAR DEVICE ENTRY FLAGS                                1189.000
                                                ;;                                                                  1190.000

                                                                                                                    1191.000
0131 6 0 0 1 1 E B 4 02 0 7F0 0 0 0 0 0 0 0 0          FR(ALEVEL)=@007F0000; INITIALIZE CURRENT ACTIVE INT LEVEL =7F   1192.000

0132 0 0 0 0 0 A F 0 02 0 0F7 0 0 0 0 0 0 0 0          T=@0F000000:T, CLRS; CLEAR S REG (INITIAL SCRATCH ADDR =0)      1193.000

0133 0 0 0 2 0 A F 0 02 0 7F0 0 0 0 0 0 0 0 0          T=@00007F00:T;                                                  1194.000

0134 6 0 0 3 0 A 6 7 02 0 FF0 0 0 0 0 0 0 0 0          FR(SCR.MASK)=@000000FF:T;  BUILD DVC ENTRY MASK =@0F7F7FFF      1195.000

0135 0 0 0 0 1 E C 0 02 D 7F0 0 0 0 0 0 0 0 0          NU=@7F000000, OTHERBANK; SET SCRATCH PAD CLEAR COUNT FOR 00-7F  1196.000

0136 0 5 7 0 1 E 8 0 0C 0 F55 0 0 0 0 0 0 0 0  155     R(SCR.ADDR)=S, RSTRHF, *LINK CLR.SCRFLGS; GO CLEAR 00 THRU 7F   1197.000

                                                                                                                    1198.000
                                                ;;                                                                  1199.000
                                                ;;          CLEAR TEMPORARY WORK CELLS                              1200.000
                                                ;;                                                                  1201.000

                                                                                                                    1202.000
0137 6 0 0 3 1 E B 7 02 0 000 0 0 0 0 0 0 0 0          FR(SCR.MASK)=@00000000; SET MASK FOR TOTAL CLEAR                1203.000

0138 0 0 0 3 1 E C 0 02 D 000 0 0 0 0 0 0 0 0          NU=@00000000, OTHERBANK; LOAD COUNT TO CLEAR 85                 1204.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SYSTEM RESET

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0139 6 0 0 1 1 E B 7 02 6 850 0 0 0 0 0 0 0 0          FR(SCR.ADDR)=@00850000, SDEST; SET SCRATCH STARTING ADDR =85      1205.000
013A 0 5 7 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0  155    *LINK CLR.SCRFLGS; GO CLEAR 85                                    1206.000
013B 0 0 0 0 1 E C 0 02 0 030 0 0 0 0 0 0 0 0          NU=@03000000; LOAD COUNT TO CLEAR 8F THRU 91                      1207.000
013C 6 0 0 1 1 E B 7 02 6 8E0 0 0 0 0 0 0 0 0          FR(SCR.ADDR)=@008E0000, SDEST; SET STARTING ADDR =8F              1208.000
013D 0 5 7 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0  155    *LINK CLR.SCRFLGS;  GO CLEAR 8E THRU 91                           1209.000
                                                                                                                       1210.000
                                    **                                                                                 1211.000
                                    **          CLEAR INTERRUPT ENTRY FLAGS AND BUILD SI VECTOR ADDRESSES              1212.000
                                    **                                                                                 1213.000
                                                                                                                       1214.000
013E 6 0 0 1 1 E B 7 02 0 FF0 0 0 0 0 0 0 0 0          FR(SCR.MASK)=@00FF0000; SET INTERRUPT ENTRY MASK                   1215.000
013F 0 0 0 3 1 E 6 0 02 D E80 0 0 0 0 0 0 0 0          DT=@000000E8, OTHERBANK; SET VECTOR ADDRESS FOR INT LEVEL 12      1216.000
0140 0 0 0 0 1 E C 0 02 0 010 0 0 0 0 0 0 0 0          NU=@01000000; SET COUNT TO BUILD LEVEL 12 & 13 VECTORS            1217.000
0141 0 5 7 0 1 0 0 0 00 C 055 0 0 0 0 0 0 0 0  155    SETRMF, *LINK CLR.SCRFLGS; GO BUILD LEVEL 12 & 13 (E8 & EC)       1218.000
0142 6 0 0 1 1 E B 7 02 6 800 0 0 0 0 0 0 0 0          FR(SCR.ADDR)=@00800000, SDEST; SET SCRATCH ADDR FOR LEVEL 00      1219.000
0143 0 5 7 0 0 0 C 0 00 F 055 0 0 0 0 0 0 0 0  155    NU=T(ZE), *LINK CLR.SCRFLGS; GO BUILD LEVEL 00 (F0)               1220.000
0144 0 0 0 3 1 E 6 0 02 0 E80 0 0 0 0 0 0 0 0          DT=@000000E8, CLDNU; SET VECTOR ADDR FOR LEVEL 01                 1221.000
0145 0 5 7 0 0 0 C 0 00 F 055 0 0 0 0 0 0 0 0  155    NU=T(ZE), *LINK CLR.SCRFLGS; GO BUILD LEVEL 01 (F8)               1222.000
0146 0 0 0 3 3 3 6 0 02 0 440 0 0 0 0 0 0 0 0          DT=@00000044+DI; BUILD VECTOR ADDR FOR LEVELS 14 THRU 23 (140)    1223.000
0147 6 0 0 1 1 E B 7 02 6 940 0 0 0 0 0 0 0 0          FR(SCR.ADDR)=@00940000, SDEST; SET LEVEL 14 SCRATCH ADDR          1224.000
0148 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0          NU=@0F000000; SET COUNT FOR LEVELS 14 THRU 23                     1225.000
0149 0 5 7 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0  155    *LINK CLR.SCRFLGS; GO BUILD LEVELS 14 THRU 23 (140 THRU 17C)      1226.000
014A 0 0 0 3 3 3 6 0 02 0 100 0 0 0 0 0 0 0 0          DT=@00000010+DI; BUILD VECTOR ADDR OF 190                         1227.000
014B 0 0 0 0 1 E C 0 02 0 5B0 0 0 0 0 0 0 0 0          NU=@5B000000; SET COUNT FOR LEVELS 24 THRU 7F                     1228.000
014C 0 5 7 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0  155    *LINK CLR.SCRFLGS; GO BUILD LEVELS 24 THRU 7F (190 THRU 2FC)      1229.000
014D 0 0 0 0 1 E F 0 02 0 807 0 0 0 0 0 0 0 0          T=@80000000,CLRS;       HALT BIT FOR TRACE                        1230.000
014E 0 0 0 0 1 E 2 0 00 0 501 0 0 0 0 0 0 0 0          PC=S, SETCC(S); CLEAR PC AND CONDITION CODES                      1231.000
014F 0 0 0 0 0 0 8 0 00 0 F00 0 0 0 0 0 0 0 0          R(TRACE)=T; SET HALT BIT IN R(TRACE)                              1232.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SYSTEM RESET

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0150 0 0 0 0 1 E 8 0 0C 8 F0D 0 0 0 0 0 0 0 0          R(ZERO)=S,RSTRHF,PCTOMAR,CLDNU; SET R(ZERO) = 0                1233.000

                                                    RESET.MAP                                                      1234.000

0151 0 6 7 0 1 E 5 0 00 D CB1 0 0 0 0 0 0 0 0  CB1     I1=S, OTHERBANK, *LINK LOAD.PSEUDO.MAP; GO INITIALIZE MAP REGS  1235.000

0152 0 0 0 0 1 E E 0 00 0 906 0 0 0 0 0 0 0 0          FULLMAR=S, SET(UNBLOCK); CLEAR MAR AND ALLOW INTERRUPTS        1236.000

0153 0 6 7 0 1 0 0 0 00 0 29F 0 0 0 0 0 0 0 0  29F     *LINK INITIALIZE.CPU.MODE;                                     1237.000

0154 0 6 0 0 1 0 0 0 00 0 A4A 0 0 0 0 0 0 0 0  A4A     *GO TO NOT.WCS.EXIT.SYS.RESET;                                 1238.000

                                                                                                                   1239.000
                                                    **                                                             1240.000
                                                    **       CLEAR SCRATCH PAD                                     1241.000
                                                    **                                                             1242.000
                                                    **       ENTRY PARAMETERS :                                    1243.000
                                                    **         R(SCR.MASK) = SCRATCH PAD CONTENTS MASK             1244.000
                                                    **         R(SCR.ADDR) = SCRATCH PAD INITIAL ADDRESS           1245.000
                                                    **         N = SCRATCH PAD CLEAR COUNT                         1246.000
                                                    **         DI = SI VECTOR ADDRESS                              1247.000
                                                    **         IF RHFLAG =0, SKIP VECTOR ADDRESS MERGE             1248.000
                                                    **         IF RHFLAG =1, EXECUTE VECTOR ADDRESS MERGE (MERGE DI WITH  1249.000
                                                    **                       ACTUAL INTERRUPT ENTRY).              1250.000
                                                    **                                                             1251.000

                                                                                                                   1252.000

                                                    CLR.SCRFLGS                                                    1253.000

0155 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0          S=SCRATCH(S);  FETCH SCRATCH PAD ENTRY                         1254.000

0156 6 4 0 0 4 0 F 7 00 D A08 0 0 0 0 0 0 0 0  158     T=S&FR(SCR.MASK), OTHERBANK, IF RHFLAG *GO TO BUILD.VECTOR;    1255.000

                                                    RESTORE.SCRATCH                                                1256.000

0157 0 0 0 4 0 0 4 0 00 A F00 0 0 0 0 0 0 0 0          SCRATCH(R(SCR.ADDR))=T;  UPDTE SCRATCH PAD ENTRY               1257.000

0158 6 0 0 1 4 3 B 7 02 6 010 0 0 0 0 0 0 0 0          FR(SCR.ADDR)=@00010000+FR(SCR.ADDR), SDEST; BUMP SCRATCH ADDR  1258.000

0159 A 4 3 0 1 0 0 0 00 0 905 0 0 0 0 0 0 0 0  155     NOD=N, DECRN, IF %NCTRZ *GO TO CLR.SCRFLGS; LOOP CLEAR SCRATCH  1259.000

                                                    DUD.1XX                                                        1260.000

015A 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *JUMPJ;  RETURN                                                1261.000

                                                    BUILD.VECTOR                                                   1262.000

015B 0 0 0 0 3 0 1 0 00 0 000 0 0 0 0 0 0 0 0          S=DI;  TRANSFER VECTOR ADDRESS                                 1263.000

015C 0 0 0 3 3 3 6 0 02 0 040 0 0 0 0 0 0 0 0          DI=@00000004+DI;  BUMP VECTOR ADDRESS FOR NEXT ENTRY           1264.000

015D 0 4 0 0 0 A F 0 00 0 007 0 0 0 0 0 0 0 0  157     T=S:T, *GO TO RESTORE.SCRATCH; COMBINE INT ENTRY AND VECTOR ADDR 1265.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SFL 32/75 CPU                                                    IPL
PC  T S M A B + D R  Y X PCH + R D M + R A D  ADDR

                                                                                                                    1266.000

                                                                 IPL                                                1267.000

015E 0 5 7 0 0 1 C 0 11 F 024 0 0 0 0 0 0 0 0  124              NU=%T(ZE),RESETFF,*LINK SFT.SYSTEM.CONSTANTS;       1268.000

015F 6 0 0 3 1 E B 1 02 0 000 0 0 0 0 0 0 0 0                    FR(RDEV)=@00000000;  SET PANEL PHYSICAL ADDR       1269.000

0160 0 0 0 0 0 0 E 0 00 D 804 0 0 0 0 0 0 0 0                    FULLMAR=T, SET(HIREG), OTHERBANK;                  1270.000

0161 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0                    SET(ENAUORD);                                      1271.000

0162 9 4 0 0 1 0 0 0 14 1 F44 0 0 0 0 0 0 0 0  164              RUN, UPACK, IF %SERIAL.PANEL *HOP $+2;             1272.000

0163 0 6 0 0 1 0 0 0 00 0 C70 0 0 0 0 0 0 0 0  C70              *GO TO S.P.IPL;                                    1273.000

0164 0 0 0 3 1 E F 0 02 0 065 0 0 0 0 0 0 0 0                    T=@00000006, RESET(ENAUORD); SET PANEL IPL ADDR REQUEST 1274.000

0165 0 6 7 0 0 0 0 0 00 0 33B 0 0 0 0 0 0 0 0  33B              *LINK RETRY.ARSTX, NOD=T;  DO ARSTX RSTX           1275.000

0166 9 4 0 0 3 0 F 0 00 0 56B 0 0 0 0 0 0 0 0  16B              T=DT, IF %IONORESP:IOTIMEOUT *HOP $+2;             1276.000

0167 0 0 0 2 1 E F 0 02 0 010 0 0 0 0 0 0 0 0                    T=@00000100; SET DEFAULT IPL ADDRESS               1277.000

                                                                 SAVE.IPL.ADDR                                      1278.000

0168 8 4 0 0 0 0 8 0 00 6 90A 0 0 0 0 0 0 0 0  16A              R(RDEV)=T, SDEST, IF ALUZ *GO TO $+2; SAVE IPL ADDR 1279.000

                                                                 SAVE.DEFAULT.IPL                                   1280.000

0169 0 4 0 1 0 0 4 0 02 A 84C 0 0 0 0 0 0 0 0  16C              SCRATCH(@84)=T, *GO TO $+3;  UPDATE DEFAULT IPL DVC ADDR 1281.000

                                                                 FETCH.DEFAULT.IPL                                  1282.000

016A 0 0 0 1 1 0 1 0 02 A 840 0 0 0 0 0 0 0 0                    S=SCRATCH(@84); FETCH DEFAULT IPL DVC ADDR         1283.000

016B 0 0 0 0 1 E 8 0 00 0 900 0 0 0 0 0 0 0 0                    R(RDEV)=S;    SAVE IPL DVC ADDR                    1284.000

016C 0 6 7 0 1 E 6 0 00 0 297 0 0 0 0 0 0 0 0  297              DT=S, *LINK IPL.CPU.MODE;                          1285.000

016D 0 0 0 2 1 E F 0 02 0 070 0 0 0 0 0 0 0 0                    T=@00000700;                                       1286.000

016E 0 0 0 1 0 0 4 0 02 A 820 0 0 0 0 0 0 0 0                    SCRATCH(@82)=T; SAVE IOCD MAIN MEMORY ADDR         1287.000

016F 0 0 0 3 0 A F 0 02 0 844 0 0 0 0 0 0 0 0                    T=@00000084:T, SET(HIREG);                         1288.000

0170 0 0 0 1 0 0 4 0 02 A 830 0 0 0 0 0 0 0 0                    SCRATCH(@83)=T; SAVE MASTER PROCESS LIST MEMORY ADDRESS 1289.000

0171 6 0 0 0 4 A B 7 02 D 120 0 0 0 0 0 0 0 0                    FR(TRACE)=@12000000:FR(TRACE),OTHERBANK; SET IPL&1ST ACCESS 1290.000

0172 6 0 0 1 1 E B 5 02 D 8E0 0 0 0 0 0 0 0 0                    FR(INTRLOC)=@008E0000,OTHERBANK; SET DUMMY SCRATCH INTR LOC 1291.000

0173 6 0 0 3 1 E B 3 02 0 000 0 0 0 0 0 0 0 0                    FR(INTRTAB)=0;                                     1292.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU IPL

```
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR
0174 0 0 0 0 1 0 0 0 00 0 F05 0 0 0 0 0 0 0 0          SET(FLAG); FLAG IDENTIFY DEVICE IN PROGRESS                    1293.000
0175 0 0 0 3 1 E F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004; SET IDENT DVC REQUEST CODE (DVC STATUS REQ)        1294.000
0176 0 5 7 0 4 0 E 0 00 0 985 0 0 0 0 0 0 0 0  185     FULLMAR=R(RDEV), *LINK FIRST.IPL.IO; GO REQUEST DVC TYPE       1295.000
                                                  CLASS.012DE.IPL.IOCD                                            1296.000
0177 0 0 0 0 1 E F 0 02 0 020 0 0 0 0 0 0 0 0          T=@02000000;                                                   1297.000
0178 0 0 0 3 0 A F 0 02 0 FF0 0 0 0 0 0 0 0 0          T=@000000FF:T;                                                 1298.000
0179 6 0 0 2 0 A B 4 02 0 7F0 0 0 0 0 0 0 0 0          FR(B.IOCD1)=@00007F00:T; BUILD IOCD1 = @02007FFF               1299.000
017A 0 0 0 0 0 0 8 0 00 0 E07 0 0 0 0 0 0 0 0          R(ICL.IOCD1)=T, CLRS; SAVE INITIAL IPL/ICL IOCD WORD 1          1300.000
017B 0 0 0 0 1 E 8 0 00 0 800 0 0 0 0 0 0 0 0          R(ICL.IOCD2)=S;                                                1301.000
017C 0 0 0 0 1 E 8 0 00 0 200 0 0 0 0 0 0 0 0          R(B.IOCD2)=S; SAVE IPL/ICL IOCD WORD 2                         1302.000
017D 0 5 0 0 1 0 0 0 00 0 093 0 0 0 0 0 0 0 0  193     *GO TO IPL.NEXT;                                               1303.000
                                                  IPL.DELIMITER                                                   1304.000
017E 0 0 3 4 1 4 8 0 00 0 000 0 0 0 0 0 0 0 0          DECRN,R(TMP0)=R(TMP0)-1;                                       1305.000
017F 2 5 0 0 1 0 0 0 00 0 BF0 0 0 0 0 0 0 0 0  1F0     IF NCTR4 *GO TO IPL.FLD1;                                      1306.000
0180 0 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  189     *GO TO IPL.HALT;                                               1307.000
                                                  IPL.DEV                                                         1308.000
0181 0 6 7 0 0 0 C 0 00 F 392 0 0 0 0 0 0 0 0  392     NU=T(7E),*LINK TEST.RAM.LOAD;                                  1309.000
0182 0 0 0 0 4 0 F 0 00 0 700 0 0 0 0 0 0 0 0          T=R(BUSREQ); RESTORE ORGINAL BUS REQUEST (WDOT-IPL)             1310.000
0183 1 4 0 0 1 0 0 0 00 0 AB1 0 0 0 0 0 0 0 0  181     IF IORETRY:IOCHBUSY *GO TO IPL.DEV;                            1311.000
0184 A 4 0 0 1 0 0 0 00 0 909 0 0 0 0 0 0 0 0  189     IF %NCTR2 *GO TO IPL.HALT;                                     1312.000
                                                  FIRST.IPL.IO                                                    1313.000
0185 0 6 7 0 1 0 0 0 00 0 337 0 0 0 0 0 0 0 0  337     *LINK IO.SEQ;                                                  1314.000
0186 1 4 0 0 1 0 0 0 00 0 AB5 0 0 0 0 0 0 0 0  185     IF IORETRY:IOCHBUSY *GO TO FIRST.IPL.IO;                       1315.000
0187 9 4 0 0 3 0 0 0 00 0 5B8 0 0 0 0 0 0 0 0  1B8     NOD=DI,IF %IONORESP:IOTIMEOUT *HOP IPL.CONTINUE;               1316.000
0188 0 0 0 0 1 0 0 0 15 0 000 0 0 0 0 0 0 0 0          CLRTO;                                                         1317.000
                                                  IPL.HALT                                                        1318.000
0189 0 0 0 2 1 E 1 0 02 0 040 0 0 0 0 0 0 0 0          S=IPL.ERR.FLG;  SET CPU STATUS FLAG                            1319.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

S F L 3 2 / 7 5 C P U I P L

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
018A 0 6 0 0 1 0 0 0 00 0 589 0 0 0 0 0 0 0 0  589        *GO TO M.55.TRAP.HALT;                                                    1320.000
                                                          IPL.CONTINUE                                                              1321.000
018B 9 4 0 0 3 0 0 0 00 0 0ED 0 0 0 0 0 0 0 0  18D        NOD=DI, IF %FLAG *HOP $+2; TEST FOR 1ST ACCESS OR CLASS F DVC             1322.000
018C 0 6 0 0 3 0 0 0 00 0 EFD 0 0 0 0 0 0 0 0  EFD        NOD=DI, *GO TO CHECK.CLASSF.IPL;                                          1323.000
018D 3 1 0 0 1 0 0 0 00 8 000 0 0 0 0 0 0 0 0              IF BMUX16 *JUMPJ; EXIT IF NO BUSY STATUS FLAG                             1324.000
018E 0 4 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0  181        *GO TO IPL.DEV;                                                           1325.000
                                                          IPL.MACRO.START                                                           1326.000
018F 0 5 7 0 0 1 C 0 11 F 024 0 0 0 0 0 0 0 0  124        NU=%T(ZE),RESETFF,*LINK SET.SYSTEM.CONSTANTS;                             1327.000
0190 0 6 7 0 1 0 0 0 00 0 26F 0 0 0 0 0 0 0 0  26F        *LINK PP.CSW.ADDR;                                                        1328.000
0191 0 6 7 0 1 0 0 0 00 0 267 0 0 0 0 0 0 0 0  267        *LINK PP.CSW.ADDR1;                                                       1329.000
0192 0 6 0 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0   F0        *GO TO FILL.PIPELINE;                                                     1330.000
                                                          IPL.NEXT                                                                  1331.000
0193 0 0 0 0 4 0 F 0 00 0 F00 0 0 0 0 0 0 0 0              T=R(ICL.IOCD1); RESTORE ICL IOCD WORD 1                                   1332.000
0194 0 0 0 0 4 0 1 0 00 0 A00 0 0 0 0 0 0 0 0              S=R(ICL.IOCD2);   RESTORE ICL IOCD WORD 2                                 1333.000
0195 0 6 7 0 1 E E 0 02 0 300 0 0 0 0 0 0 0 0  300        FULLMAR=(MEMORY.WRITE↑20&@FF000000),*LINK MEMORY.WRITE;CLEAR MAR          1334.000
0196 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0              FULLMAR=4+MAR;                                                            1335.000
0197 0 6 7 0 1 E F 0 00 0 300 0 0 0 0 0 0 0 0  300        T=S, *LINK MEMORY.WRITE;                                                  1336.000
0198 0 0 0 0 4 0 E 0 00 0 900 0 0 0 0 0 0 0 0              FULLMAR=R(RDEV);                                                          1337.000
0199 0 0 0 0 1 E F 0 02 0 900 0 0 0 0 0 0 0 0              T=@90000000;     WDOT REQ START IO AND IPL                                1338.000
019A 0 5 7 0 1 0 0 0 00 0 081 0 0 0 0 0 0 0 0  181        *LINK IPL.DEV;                                                            1339.000
019B 0 0 0 3 1 E F 0 02 0 080 0 0 0 0 0 0 0 0              T=@00000008;                                                              1340.000
019C 0 5 7 0 1 0 0 0 00 0 084 0 0 0 0 0 0 0 0  184        *LINK IPL.DEV+3;                                                          1341.000
019D 0 0 0 3 1 E E 0 02 0 000 0 0 0 0 0 0 0 0              FULLMAR=0;                                                                1342.000
019E 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B        *LINK MEMORY.READ;                                                        1343.000
019F 0 0 0 0 3 0 0 0 00 0 000 0 0 0 0 0 0 0 0              NOD=DI;   TEST FOR FIRST WORD EQUAL TO ZERO                               1344.000
01A0 0 0 0 0 3 D F 0 02 0 FF0 0 0 0 0 0 0 0 0              T=@FF000000&DI; GET FIRST BYTE OF FIRST WORD IN T                         1345.000
01A1 C 4 0 0 4 0 0 0 00 D E0B 0 0 0 0 0 0 0 0  1AB        NOD=R(IPL.FLAGS), OTHERBANK, IF NALUZ *GO TO IPL.SCAN;   EXIT IF          1346.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    IPL
PC   TSMAB+DR YX PCH+BDM+BAD ADDR

                                                *                          THE FIRST WORD OF THE FIRST IPL RECORD      1347.000
                                                *                          IS NOT ZERO.                                1348.000
01A2 5 5 0 0 1 0 0 0 00 4 08F 0 0 0 0 0 0 0 0 18F       IF %BMUX00 *GO TO IPL.MACRO.START; EXIT IF RECORD COUNT > 1     1349.000
                                                **  HANDLE FIRST ICL RECORD OF A IOCD FOR SUBSEQUENT ICL            1350.000
                                                **  (FIRST WORD OF FIRST RECORD = 0)                                1351.000
01A3 6 0 0 0 1 E B 6 02 0 F00 0 0 0 0 0 0 0 0           FR(IPL.FLAGS)=@F0000000;  SET IOCD RECEIVED FLAGS               1352.000
01A4 0 0 0 3 1 E E 0 02 D 040 0 0 0 0 0 0 0 0           FULLMAR=4, OTHERBANK;                                           1353.000
01A5 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0 30B       *LINK MEMORY.READ;                                              1354.000
01A6 0 0 0 0 3 0 8 0 00 0 F00 0 0 0 0 0 0 0 0           R(ICL.IOCD1)=DI;                                                1355.000
01A7 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0           FULLMAR=4+MAR;                                                  1356.000
01A8 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0 30B       *LINK MEMORY.READ;                                              1357.000
01A9 0 0 0 0 3 0 8 0 00 0 800 0 0 0 0 0 0 0 0           R(ICL.IOCD2)=DI;                                                1358.000
01AA 0 5 0 0 1 0 0 0 00 0 093 0 0 0 0 0 0 0 0 193       *GO TO IPL.NEXT;                                                1359.000
                                                IPL.SCAN                                                                1360.000
01AB 0 0 0 0 3 D F 0 02 0 FF0 0 0 0 0 0 0 0 0           T=@FF000000&DI;                                                 1361.000
01AC 0 0 0 0 0 5 0 0 02 0 2A0 0 0 0 0 0 0 0 0           NOD=@2A000000-T;        TEST FOR *                              1362.000
01AD 0 0 0 1 3 D F 0 02 0 FF0 0 0 0 0 0 0 0 0           T=@00FF0000&DI;                                                 1363.000
01AE C 5 0 0 1 0 0 0 00 0 08F 0 0 0 0 0 0 0 0 18F       IF NALUZ *GO TO IPL.MACRO.START; EXIT IF NO * IN BYTE 1         1364.000
01AF 0 0 0 1 0 5 0 0 02 0 450 0 0 0 0 0 0 0 0           NOD=@00450000-T;        TEST FOR E                              1365.000
01B0 0 0 0 1 0 5 0 0 02 0 490 0 0 0 0 0 0 0 0           NOD=@00490000-T;          TEST FOR I                            1366.000
01B1 B 4 0 1 0 5 0 0 02 0 445 0 0 0 0 0 0 0 0 185       NOD=@00440000-T,IF ALUZ *GO TO IPL.END; EXIT BYTE 2 = E          1367.000
01B2 B 5 0 0 4 0 0 0 00 0 EC9 0 0 0 0 0 0 0 0 1C9       NOD=R(IPL.FLAGS), OTHERBANK, IF ALUZ *GO TO IPL.INTR;           1368.000
                                                *                              : EXIT IF BYTE #2 = I                    1369.000
01B3 B 4 0 0 4 0 0 0 00 D F07 0 0 0 0 0 0 0 0 1B7       NOD=R(IPL.FLAGS), OTHERBANK, IF ALUZ *GO TO IPL.EDEV;           1370.000
                                                *                              ; EXIT IF BYTE #2 = D                    1371.000
01B4 0 5 0 0 1 0 0 0 00 0 08F 0 0 0 0 0 0 0 0 18F       *GO TO IPL.MACRO.START; EXIT BYTE 2 NOT EQUAL TO E, D, OR I     1372.000
                                                IPL.END                                                                 1373.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                                    IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

01B5 0 0 0 0 4 0 1 0 00 0 200 0 0 0 0 0 0 0 0          S=R(B.IOCD2);     RESTORE READ BINARY IOCD WORD 2                  1374.000
01B6 0 5 0 0 4 0 F 0 00 0 C95 0 0 0 0 0 0 0 0  195     T=R(B.IOCD1), *GO TO IPL.NEXT+2; RESTORE READ.BINARY IOCD WORD 1 1375.000
                                                     IPL.EDEV                                                         1376.000
01B7 3 5 7 0 1 0 0 0 00 8 0F2 0 0 0 0 0 0 0 0  1F2     IF BMUX16 *LINK IPL.CLEAR;                                       1377.000
01B8 0 5 7 0 1 0 0 0 00 0 0D0 0 0 0 0 0 0 0 0  1D0     *LINK IPL.READ.ENTRY;                                            1378.000
                                                     **            WRITE 86 DEVICE INTERRUPT ENTRY                    1379.000
01B9 0 0 0 0 4 0 F 0 00 0 600 0 0 0 0 0 0 0 0          T=R(TMP6);                                                       1380.000
01BA 0 0 0 1 0 A 1 0 02 0 800 0 0 0 0 0 0 0 0          S=@00800000:T;                                                   1381.000
01BB 0 0 0 0 5 0 4 0 00 A 500 0 0 0 0 0 0 0 0          SCRATCH(S)=R(TMP5,HWS);          WRITE DEV INTR ENTRY            1382.000
                                                     **            COMPLEMENT DEVICE INTERRUPT LEVEL                  1383.000
01BC 0 0 0 0 4 0 F 0 00 0 600 0 0 0 0 0 0 0 0          T=R(TMP6);               LOAD DEVICE ENTRY                       1384.000
01BD 0 0 0 5 0 D 1 0 02 0 000 0 0 0 0 0 0 0 0          S=@FF00FFFF&T;           CLEAR INTERRUPT LEVEL                   1385.000
01BE 0 0 0 1 0 C F 0 02 0 FF0 0 0 0 0 0 0 0 0          T=@00FF0000&%T;          ISOLATE COMPLEMENT INTERRUPT LEVEL       1386.000
01BF 0 5 7 0 0 A 8 0 00 0 65A 0 0 0 0 0 0 0 0  15A     R(TMP6)=S:T,*LINK DUD.1XX;                                       1387.000
                                                     **            WRITE FIRST 86 DEVICE ENTRY                        1388.000
01C0 0 0 0 0 5 0 1 0 00 0 500 0 0 0 0 0 0 0 0          S=R(TMP5,HWS);           LOAD 86 DEVICE ADDRESS                  1389.000
01C1 0 0 0 0 5 0 D 0 00 0 700 0 0 0 0 0 0 0 0          NL=R(TMP7,HWS);          LOAD NUMBER OF SUBADDRESSES             1390.000
01C2 0 0 0 0 4 0 4 0 00 A 600 0 0 0 0 0 0 0 0          SCRATCH(S)=R(TMP6);              WRITE DEVICE ENTRY              1391.000
01C3 2 5 3 0 1 0 0 0 00 0 993 0 0 0 0 0 0 0 0  193     DECRN,IF NCTR7 *GO TO IPL.NEXT;                                  1392.000
                                                     IPL.DEV.LOAD                                                     1393.000
                                                     **            WRITE SUBSEQUENT 86 DEVICE ENTRIES                 1394.000
01C4 2 5 3 0 1 0 0 0 00 0 993 0 0 0 0 0 0 0 0  193     DECRN,IF NCTR7 *GO TO IPL.NEXT;                                  1395.000
01C5 0 0 0 4 1 6 8 0 00 0 600 0 0 0 0 0 0 0 0          R(TMP6)=R(TMP6)+1;       INCREMENT SUB ADDRESS                   1396.000
01C6 0 0 0 1 1 E 6 0 02 0 010 0 0 0 0 0 0 0 0          DI=@00010000;                                                    1397.000
01C7 0 0 0 0 3 3 1 0 00 0 000 0 0 0 0 0 0 0 0          S=S+DI;                  INCREMENT ENTRY ADDRESS                 1398.000
01C8 0 4 0 0 4 0 4 0 00 A 604 0 0 0 0 0 0 0 0  1C4     SCRATCH(S)=R(TMP6),*GO TO IPL.DEV.LOAD;                          1399.000
                                                     IPL.INTR                                                         1400.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
SEL  32/75  CPU                                                    IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

01C9 3 5 7 0 1 0 0 0 00 8 0F2 0 0 0 0 0 0 0 0  1F2        IF BMUX16 *LINK IPL.CLEAR;                                  1401.000

01CA 0 5 7 0 1 0 0 0 00 0 0D0 0 0 0 0 0 0 0 0  1D0        *LINK IPL.READ.ENTRY;                                       1402.000

                                                  **              WRITE 86 INTERRUPT ENTRY                          1403.000

01CB 0 0 0 1 1 E 1 0 02 0 800 0 0 0 0 0 0 0 0             S=@00800000;                                                1404.000

01CC 0 0 0 0 5 A 1 0 00 0 500 0 0 0 0 0 0 0 0             S=S:R(TMP5,HWS);          SET INTERRUPT ENTRY OFFSET        1405.000

01CD 0 0 0 0 5 0 F 0 00 0 600 0 0 0 0 0 0 0 0             T=R(TMP6,HWS);                                              1406.000

01CE 0 0 0 1 0 A F 0 02 0 800 0 0 0 0 0 0 0 0             T=@00800000:T;            SET RTOM BIT                      1407.000

01CF 0 5 0 0 0 0 4 0 00 A 093 0 0 0 0 0 0 0 0  193        SCRATCH(S)=T,*GO TO IPL.NEXT;                               1408.000

                                                                                                                      1409.000
                                                  **                                                                  1410.000
                                                  **      ASCII STRING PROCESSOR                                      1411.000
                                                  **                                                                  1412.000
                                                  **              CONVERT ASCII TO HEX                                1413.000
                                                  **                      OPERAND 1 RETURNED IN REG 5                 1414.000
                                                  **                      OPERAND 2 RETURNED IN REG 6                 1415.000
                                                  **                      OPERAND 3 RETURNED IN REG 7                 1416.000
                                                  **              DELIMITERS                                          1417.000
                                                  **                      = OPERAND 1 DELIMITER                       1418.000
                                                  **                      , OPERAND 2 DELIMITER                       1419.000
                                                  **                     SP ANY OPERAND AND STRING DELIMITER          1420.000
                                                  **                                                                  1421.000

                                                                                                                      1422.000

                                                  IPL.READ.ENTRY                                                      1423.000

01D0 0 0 0 0 1 E C 0 02 0 0A7 0 0 0 0 0 0 0 0             MH=@0A000000,CLRS;                                          1424.000

01D1 0 0 0 0 1 E 8 0 00 0 500 0 0 0 0 0 0 0 0             R(TMP5)=S;                CLEAR OPERAND 1 RETURN REG        1425.000

01D2 0 0 0 0 0 0 8 0 00 0 600 0 0 0 0 0 0 0 0             R(TMP6)=T;                CLEAR OPERAND 2 RETURN REG        1426.000

01D3 0 0 0 0 0 0 8 0 00 0 700 0 0 0 0 0 0 0 0             R(TMP7)=T;                CLEAR OPERAND 3 RETURN REG        1427.000

                                                  IPL.NEXT.WORD                                                       1428.000

01D4 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0             FULLMAR=@00000004+MAR;                                      1429.000

01D5 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B        *LINK MEMORY.READ;                                          1430.000

01D6 0 0 0 3 1 E 8 0 02 0 040 0 0 0 0 0 0 0 0             R(TMP0)=@00000004;                                          1431.000

                                                  IPL.NEXT.BYTE                                                       1432.000

01D7 0 0 0 0 3 0 F 0 00 0 007 0 0 0 0 0 0 0 0             T=DI,CLRS;                                                  1433.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                                                                    IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

01D8 0 0 0 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0          S=SNIBL,TNIBL;                                                    1434.000
01D9 0 0 0 2 1 E 6 0 00 2 000 0 0 0 0 0 0 0 0          DI=SNIBL,TNIBL;                                                   1435.000
01DA 0 0 0 0 0 0 8 0 00 0 300 0 0 0 0 0 0 0 0          R(TMP3)=T;                                                        1436.000
01DB 0 0 0 3 3 5 0 0 02 0 200 0 0 0 0 0 0 0 0          NOD=@00000020-DI;          TEST FOR SPACE                         1437.000
01DC 0 0 0 3 3 5 0 0 02 0 3D0 0 0 0 0 0 0 0 0          NOD=@0000003D-DI;          TEST FOR =                             1438.000
01DD B 1 0 3 3 5 0 0 02 0 2C0 0 0 0 0 0 0 0 0          NOD=@0000002C-DI,IF ALU7 *JUMPJ;          TEST FOR ,          1439.000
01DE B 5 0 0 1 0 0 0 00 0 07E 0 0 0 0 0 0 0 0  17E     IF ALUZ *GO TO IPL.DELIMITER;                                     1440.000
01DF B 5 0 0 1 0 0 0 00 0 07E 0 0 0 0 0 0 0 0  17E     IF ALUZ *GO TO IPL.DELIMITER;                                     1441.000
01E0 0 0 0 3 3 5 0 0 02 0 2F0 0 0 0 0 0 0 0 0          NOD=@0000002F-DI;      TEST 0-F LOWER LIMITS                      1442.000
01E1 0 0 0 3 3 5 0 0 02 0 460 0 0 0 0 0 0 0 0          NOD=@00000046-DI;      TEST 0-F UPPER LIMITS                      1443.000
01E2 A 5 0 0 1 0 0 0 00 0 889 0 0 0 0 0 0 0 0  189     IF %ALUNEG *GO TO IPL.HALT;                                       1444.000
01E3 2 5 0 0 1 0 0 0 00 0 889 0 0 0 0 0 0 0 0  189     IF ALUNEG *GO TO IPL.HALT;                                        1445.000
01F4 0 0 0 3 3 5 0 0 02 0 400 0 0 0 0 0 0 0 0          NOD=@00000040-DI;      TEST A-F LOWER LIMITS                      1446.000
01E5 0 0 0 3 3 5 0 0 02 0 390 0 0 0 0 0 0 0 0          NOD=@00000039-DI;      TEST 0-9 UPPER LIMITS                      1447.000
01E6 2 4 0 0 1 0 0 0 00 0 80A 0 0 0 0 0 0 0 0  1EA     IF ALUNEG *GO TO IPL.AF;                                          1448.000
01E7 2 5 0 0 1 0 0 0 00 0 889 0 0 0 0 0 0 0 0  189     IF ALUNEG *GO TO IPL.HALT;                                        1449.000
01E8 0 0 0 3 3 D 1 0 02 0 0F0 0 0 0 0 0 0 0 0          S=@0000000F&DI;                                                   1450.000
01E9 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  1EC     *GO TO IPL.FOLD;                                                  1451.000
                                                IPL.AF                                                                   1452.000
01EA 0 0 0 3 3 D 6 0 02 0 0F0 0 0 0 0 0 0 0 0          DI=@0000000F&DI;                                                  1453.000
01EB 0 0 0 3 3 3 1 0 02 0 090 0 0 0 0 0 0 0 0          S=@00000009+DI;                                                   1454.000
                                                IPL.FOLD                                                                 1455.000
01EC 0 0 0 0 4 0 F 3 00 0 000 0 0 0 0 0 0 0 0          T=R(NCTR);                                                        1456.000
01ED 0 0 0 0 1 0 0 0 00 2 000 0 0 0 0 0 0 0 0          TNIRL;                                                            1457.000
01EE 0 5 7 0 0 A B 3 00 0 05A 0 0 0 0 0 0 0 0  15A     R(NCTR)=S:T,*LINK DUD.1XX;          ADD HEX DIGIT                 1458.000
01EF 0 5 7 4 1 4 8 0 00 0 05A 0 0 0 0 0 0 0 0  15A     R(TMP0)=R(TMP0)-1,*LINK DUD.1XX;                                  1459.000
                                                IPL.FLD1                                                                 1460.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                                IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

01F0 B 5 0 0 4 0 6 0 00 0 3D4 0 0 0 0 0 0 0 0  1D4        DI=R(TMP3),IF ALU7 *GO TO IPL.NEXT.WORD;                    1461.000
01F1 0 5 0 0 1 0 0 0 00 0 0D7 0 0 0 0 0 0 0 0  1D7        *GO TO IPL.NEXT.BYTE;                                      1462.000
                                                    IPL.CLEAR                                                       1463.000
01F2 6 0 0 3 1 E B 7 02 6 004 0 0 0 0 0 0 0 0              FR(SCR.MASK)=@00000000, RESET(HIREG), SDEST; SET MASK & ADDR =0  1464.000
01F3 0 0 0 4 1 E C 0 02 0 7F0 0 0 0 0 0 0 0 0              NU=@7FFFFFFF; SET SCRATCH COUNT TO CLEAR 00-7F              1465.000
01F4 0 5 7 0 0 0 8 0 0C 0 F55 0 0 0 0 0 0 0 0  155        R(SCR.ADDR)=T, RSTRHF, *LINK CLR.SCRFLGS; GO CLEAR 00-7F    1466.000
01F5 0 0 0 1 1 E 6 0 02 C 800 0 0 0 0 0 0 0 0              DI=@00800000, SETRHF;  SET UP TO INITIALIZE RTOM INT ENTRIES 1467.000
01F6 0 4 7 0 1 E C 0 02 0 01B 0 0 0 0 0 0 0 0  1FB        NU=@01000000, *LINK DISPATCH.CLEAR; GO INITIALIZE 80 AND 81 1468.000
01F7 6 0 0 1 1 E 6 7 02 6 920 0 0 0 0 0 0 0 0              FR(SCR.ADDR)=@00920000, SDEST; SET SCRATCH ADDR FOR 92 THRU 7F 1469.000
01F8 0 4 7 0 1 E C 0 02 0 6DB 0 0 0 0 0 0 0 0  1FB        NU=@6DDDDDDD, *LINK DISPATCH.CLEAR; GO INITIALIZE 92 THRU 7F 1470.000
01F9 6 0 0 2 4 A B 6 02 0 804 0 0 0 0 0 0 0 0              FR(TPL.FLAGS)=@00008000:FR(TPL.FLAGS), SET(HIREG);          1471.000
                                                    *                                ; SET SCRATCH PAD CLEARED FLAG 1472.000
01FA 0 1 0 0 1 0 0 0 0C 0 000 0 0 0 0 0 0 0 0              RSTRHF, *JUMPJ; EXIT                                        1473.000
                                                    DISPATCH.CLEAR                                                  1474.000
01FB 0 5 0 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0  155        *GO TO CLR.SCRFLGS; GO CLEAR SCRATCH PAD                    1475.000
                                                    FORMAT.75.AEXP.75MODE                                           1476.000
01FC 9 4 0 0 1 0 0 0 00 0 F7E 0 0 0 0 0 0 0 0  1FE        IF %MAPMODE *HOP $+2;                                       1477.000
01FD 0 0 0 1 0 A F 0 02 0 400 0 0 0 0 0 0 0 0              T=@00400000:T; SET MAP MODE DISPLAY BIT                     1478.000
01FE 0 0 0 1 0 A F 0 02 0 800 0 0 0 0 0 0 0 0              T=@00800000:T;  SET 75 MODE DISPLAY BIT                     1479.000
                                                    FORMAT.75.AEXP                                                  1480.000
01FF 4 4 0 0 1 0 0 0 00 A 001 0 0 0 0 0 0 0 0  201        IF ENBL.AEXP *GO TO SET.PSD1.AEXP;                          1481.000
0200 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPJ;  EXIT                                               1482.000
                                                    SET.PSD1.AEXP                                                   1483.000
0201 0 1 0 0 0 A F 0 02 0 010 0 0 0 0 0 0 0 0              T=@01000000:T, *JUMPJ; SET EN ABLE AEXP BIT IN PSD1         1484.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                    S I P U   I N S T R U C T I O N
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1485.000
                                                    ***                                                             1487.000
                                                    ***  SECONDARY DECODE OF OP CODE = 000A (Q=00) ***               1488.000
                                                    ***                                                             1489.000
                                                                                                                    1490.000
                                                    SIPU                                                            1491.000
0202 4 6 0 0 1 0 0 0 00 2 F63 0 0 0 0 0 0 0 0  F63          IF IPU *6010 SIPU.IPU.TRAP;    SIPU INTERNAL EVENT     1492.000
                                                    *                                                               1493.000
                                                    SIPU.RETURN                                                     1494.000
                                                    *                                                               1495.000
0203 0 0 0 0 1 0 0 0 00 0 505 0 0 0 0 0 0 0 0               RESET(IPU.HALT);               CLEAR STOP FLAG         1496.000
0204 0 0 0 0 1 0 0 0 00 0 E05 0 0 0 0 0 0 0 0             SET(IPU.START);  GENERATE A TRAP SIGNAL TO THE IPU        1497.000
0205 0 0 0 0 1 0 0 0 00 0 605 0 0 0 0 0 0 0 0             RESET(IPU.START);                                         1498.000
0206 4 1 0 0 1 0 0 0 00 2 000 0 0 0 0 0 0 0 0               IF IPU *JUMPJ;               RETURN TO TRAP HANDLER     1499.000
0207 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829        *GO TO FETCH.RETURN;                                      1500.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU PROGRAMMER PANEL
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                1501.000

                                                                                                                                1502.000
                                                                *                                                               1503.000
                                                                *       PROGRAMMER PANEL CPU SUPPORT                            1504.000
                                                                *            HALT MODE                                          1505.000
                                                                *            PANEL INITIATED TRANSFERS                          1506.000

                                                                                                                                1507.000
                                                                (@20A)                                                          1508.000

                                                                PPATTN.HALT1                                                    1509.000

020A 0 0 0 0 1 0 0 0 1C 1 083 0 0 0 0 0 0 0 0                           RFAD,IGNSTOP,FRCWORD; CLEAR ADDRESS STOP FLAG           1510.000

                                                                PPATTN.HALT2                                                    1511.000

020B 9 4 0 0 1 0 0 0 00 0 F4E 0 0 0 0 0 0 0 0  20E              IF %SERIAL.PANEL *HOP $+3;                              1512.000

020C 0 6 7 0 1 E F 0 00 0 C21 0 0 0 0 0 0 0 0  C21              T=S, *LINK S.P.PSW.PD;                                  1513.000

020D 0 5 0 0 1 0 0 0 00 0 044 0 0 0 0 0 0 0 0  244              *GO TO PP.END; IF SERIAL PANEL, BYPASS PARALLEL PANEL   1514.000

020E 6 0 0 0 1 E B 1 02 0 F00 0 0 0 0 0 0 0 0                   FR(PDEV)=@F0000000;     SET PANEL IOC ADDR              1515.000

020F 0 6 7 0 1 E F 0 00 0 33R 0 0 0 0 0 0 0 0  33R              T=S,*LINK RETRY.ARSTX;                                  1516.000

0210 0 0 0 0 3 0 8 0 00 0 000 0 0 0 0 0 0 0 0                   R(TMP0)=DI; SAVE FUNCT AND SOURCE DEST FROM PANEL       1517.000

0211 9 4 0 0 1 0 0 0 00 0 5B3 0 0 0 0 0 0 0 0  213              IF %IONORESP:IOTIMEOUT *GO TO $+2;                      1518.000

0212 0 5 0 0 1 0 0 0 00 0 042 0 0 0 0 0 0 0 0  242              *GO TO PP.NPRESENT;                                     1519.000

0213 0 0 0 0 5 0 F 0 00 F 000 0 0 0 0 0 0 0 0                   T=R(TMP0,HWS,7E);  SHIFT FUNCTION TO LSB'S              1520.000

0214 0 0 0 7 0 3 0 0 12 0 FF0 0 0 0 0 0 0 0 0                   NOD=@FFFFFFFF + T, SAVESIGN;    -1 + FUNCTION           1521.000

0215 0 0 0 1 1 9 D 0 02 0 0F0 0 0 0 0 0 0 0 0                   NL=%@000F0000;  SET BIT FLAG REG NUMBER TO 15 (TRACE)   1522.000

0216 B 5 0 0 1 0 0 0 00 8 048 0 0 0 0 0 0 0 0  248              PCTOBAR, IF ALU7 *GO TO PPATTN.WT;                      1523.000

0217 3 4 0 0 1 0 0 0 00 1 00F 0 0 0 0 0 0 0 0  21E              IF SIGNSAVE *GO TO PPATTN.RD;                           1524.000

                                                                                                                                1525.000
                                                                *                                                               1526.000
                                                                *               INSTRUCTION STEP REQUESTED                      1527.000
                                                                *                                                               1528.000

                                                                                                                                1529.000
                                                                S.P.INSTR.STEP                                                  1530.000

                                                                PPINST.STEP                                                     1531.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                          P R O G R A M M E R   P A N E L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0218 0 0 0 4 4 D F 3 02 D 7F0 0 0 0 0 0 0 0 0          T=@7FFFFFFF & R(NCTR), OTHERBANK;                              1532.000

0219 0 0 0 0 0 A B 3 02 0 400 0 0 0 0 0 0 0 0          R(NCTR)=@40000000:T;                                           1533.000

021A 0 0 0 0 1 0 0 0 14 D 004 0 0 0 0 0 0 0 0          OTHERBANK, UPACK, RESET(HIREG);                                1534.000

021B 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0          SET(ENAUORD);                                                  1535.000

021C 0 0 0 0 1 0 0 0 00 1 804 0 0 0 0 0 0 0 0          RUN,SET(TRACEFF);   SET TRACEFF FF                             1536.000

021D 0 6 0 0 1 0 0 0 00 0 0BA 0 0 0 0 0 0 0 0   BA     *GO TO NPM.CHECK; GO CHECK FOR SAVED ERRORS                    1537.000

                                                                                                                    1538.000
                                                *                                                                   1539.000
                                                *              READ FUNCTION REQUESTED                              1540.000
                                                *                                                                   1541.000
                                                *         CODES  0-7 EQUAL FILE READ REQUESTS                       1542.000
                                                *         CODE   8   EQUAL PSW 1 READ REQUEST                       1543.000
                                                *         CODE   9   EQUAL READ EFFECTIVE ADDRESS                   1544.000
                                                *         CODE   A   N/A                                            1545.000
                                                *         CODE   B   EQUAL PSD 2 READ REQUEST                       1546.000
                                                *                                                                   1547.000
                                                PPATTN.RD                                                           1548.000

021E 0 0 0 3 4 D 8 0 02 0 0F0 0 0 0 0 0 0 0 0          R(TMP0)=@0000000F & R(TMP0);  NOTE: LIT & REGNO =              1549.000

021F 0 0 0 7 0 3 0 0 12 0 F80 0 0 0 0 0 0 0 0          NOD=@FFFFFFF8 + T, SAVESIGN;    S/D CODE -8                    1550.000

0220 0 0 0 7 0 3 0 0 00 0 F70 0 0 0 0 0 0 0 0          NOD=@FFFFFFF7 + T;  S/D CODE -9                                1551.000

0221 B 5 0 0 1 0 0 0 00 0 035 0 0 0 0 0 0 0 0  235     IF ALUZ *GO TO PPATTN.RD.PSW;                                  1552.000

0222 B 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  226     IF ALUZ *GO TO READ.EFFECTIVE.ADDR;                            1553.000

0223 3 5 0 0 1 0 0 0 00 1 033 0 0 0 0 0 0 0 0  233     IF SIGNSAVE *GO TO PPATTN.RD.FILE;                             1554.000

                                                PPREAD.PSD2                                                         1555.000

0224 0 0 0 1 1 0 1 0 02 A 900 0 0 0 0 0 0 0 0          S=SCRATCH(@90); FETCH PSD 2 FROM SCRATCH PAD                   1556.000

0225 0 5 0 0 1 E F 0 00 0 040 0 0 0 0 0 0 0 0  240     T=S, *GO TO PPATTN.RD.M1;  GO TRANSMIT PSD2 TO PP.PANEL         1557.000

                                                                                                                    1558.000
                                                *                                                                   1559.000
                                                *       READ EFFECTIVE ADDRESS(USED BY THE PARALLEL AND SERIAL PANELS) 1560.000
                                                *                                                                   1561.000
                                                *       ENTRY PARAMETER :                                           1562.000
                                                *         MAR = CURRENT PC                                          1563.000
                                                *                                                                   1564.000
                                                *       EXIT PARAMETER :                                            1565.000
                                                *         T = EFFECTIVE ADDRESS OF THE INSTRUCTION ADDRESSED BY PC  1566.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU PROGRAMMER PANEL

```
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                *                                                                    1567.000

                                                                                                                     1568.000

                                                READ.EFFECTIVE.ADDR                                                  1569.000

0226 0 0 0 5 1 E F 0 00 0 004 0 0 0 0 0 0 0 0          T=STATUS, RESET(HIREG);  GET CURRENT CPU STATUS               1570.000

0227 0 0 0 0 0 D C 0 02 0 040 0 0 0 0 0 0 0 0          NU=@04000000%T; TEST FOR EXTENDED INDEXING                    1571.000

0228 0 0 0 1 1 A 6 0 02 0 100 0 0 0 0 0 0 0 0          DI=@00100000:MAR; SET INDIRECT BIT IN DI                      1572.000

0229 C 4 0 4 1 E F 0 02 0 00B 0 0 0 0 0 0 0 0  228     T=@00FFFFFF, IF NALUZ *HOP $+2; MASK = @ 00FFFFFF FOR EXT.I   1573.000

022A 0 0 0 5 0 D F 0 02 0 0F0 0 0 0 0 0 0 0 0          T=@FF0FFFFF%I; MASK = @ 000FFFFF FOR NON-EXTENDED INDEXING    1574.000

022B 0 6 7 4 3 3 7 1 00 0 FEE 0 0 0 0 0 0 0 0  FEE     MARIX=R(DIX)+DI, *LINK EFFECTIVE.ADDR.INDIR;GO COMPUTE EFF. ADDR 1575.000

022C A 5 0 0 0 D F 0 00 0 930 0 0 0 0 0 0 0 0  230     T=S%T, IF %NCTR7 *GO TO RD.EFF.ERR;                           1576.000

022D 0 0 0 0 1 0 0 0 00 0 804 0 0 0 0 0 0 0 0          SET(HIREG);                                                   1577.000

022E 1 1 0 0 1 0 0 0 00 0 F40 0 0 0 0 0 0 0 0          IF SERIAL.PANEL *JUMPJ;  SERIAL PANEL EXIT                    1578.000

                                                EFFECTIVE.ADDR.EXIT                                                  1579.000

022F 0 4 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  230     *GO TO PPATTN.RD.M;                                           1580.000

                                                RD.EFF.ERR                                                           1581.000

0230 9 4 0 0 1 0 0 0 00 0 F52 0 0 0 0 0 0 0 0  232     IF %UPRNDPE *GO TO EFF.ERR.NPM;                               1582.000

0231 0 4 0 0 1 E F 0 02 0 40D 0 0 0 0 0 0 0 0  230     T=@40000000, *GO TO PPATTN.RD.M;                              1583.000

                                                EFF.ERR.NPM                                                          1584.000

0232 0 4 0 0 1 E F 0 02 0 80D 0 0 0 0 0 0 0 0  230     T=@80000000, *GO TO PPATTN.RD.M;                              1585.000

                                                                                                                     1586.000
                                                *                                                                    1587.000
                                                *                                                                    1588.000
                                                *              READ FILE REQUEST                                     1589.000
                                                *                                                                    1590.000

                                                                                                                     1591.000

                                                PPATTN.RD.FILE                                                       1592.000

0233 0 5 7 0 5 1 0 0 00 0 06F 0 0 0 0 0 0 0 0  26E     NL=%R(TMP0,HWS), *LINK DUD.2; SET ONES-COMP OF REG ADDR IN N. 1593.000

0234 0 4 0 0 4 0 F 3 00 0 00D 0 0 0 0 0 0 0 0  230     T=R(NCTR),OTHERBANK, *GO TO PPATTN.RD.M;  READ FILE           1594.000

                                                                                                                     1595.000
                                                *                                                                    1596.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L  3 2 / 7 5  C P U                                                    P R O G R A M M E R   P A N E L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                *              READ PSW REQUEST(USED BY THE PARALLEL AND SERIAL PANELS) 1597.000
                                                                *                                                                       1598.000
                                                                *       ENTRY PARAMETER :                                               1599.000
                                                                *         MAR = CURRENT PC                                              1600.000
                                                                *       EXIT PARAMETER :                                                1601.000
                                                                *         T = FORMATTED PSW (NOTE: BIT 31 IS USED TO DISLAY BLOCK MODE) 1602.000
                                                                *                                                                       1603.000

                                                                                                                                        1604.000

                                                                PPATTN.RD.PSW                                                           1605.000

0235 0 0 0 0 4 0 F 0 00 0 900 0 0 0 0 0 0 0 0                           T=R(STMASK), OTHERBANK; GET CPU STATUS MASK                      1606.000

0236 0 0 0 4 1 0 1 0 00 0 800 0 0 0 0 0 0 0 0                           S=R(PCMASK)&MAR, OTHERBANK; CLEAN CURRENT PC AND MOVE TO S       1607.000

0237 4 4 0 5 0 0 F 0 00 8 009 0 0 0 0 0 0 0 0  239                      T=STATUS&T, IF UNBLOCK *GO TO $+2;GET PRIV BIT,CC'S, & EXT BIT   1608.000

0238 0 0 0 3 0 A F 0 02 0 010 0 0 0 0 0 0 0 0                           T=@00000001:T; SET PSW BIT 31 FOR A BLOCK INT FLAG               1609.000

0239 0 0 0 0 4 0 0 0 00 0 F00 0 0 0 0 0 0 0 0                           NOD=R(TRACE), OTHERBANK; TEST HALFWORD INDICATOR                 1610.000

023A 3 4 0 0 0 A F 0 00 8 00C 0 0 0 0 0 0 0 0  23C                      T=S:T, IF BMUX16 *GO TO $+2; BRANCH IF NOT HALFWORD FLAG         1611.000

023B 0 0 0 3 0 A F 0 02 0 020 0 0 0 0 0 0 0 0                           T=@00000002:T; SET PSW HALFWORD INDICATOR BIT                    1612.000

023C 4 6 7 0 1 0 0 0 00 C 1FC 0 0 0 0 0 0 0 0  1FC                      IF MODE75S *LINK FORMAT.75.AEXP.75MODE; GO FORMAT BIT 6,7        1613.000

                                                                                                                                        1614.000

                                                                PPATTN.RD.M                                                             1615.000

023D 9 4 0 0 1 E E 0 02 0 55F 0 0 0 0 0 0 0 0  23F                      FULLMAR=@55000000,IF %OPRNDPE:OPTIMEOUT:OPNORESP *GO TO $+2;     1616.000

023E 0 0 0 0 1 0 0 0 1C 0 083 0 0 0 0 0 0 0 0                           READ,IGNSTOP;            CLEAR ERRORS                           1617.000

023F 1 1 0 0 1 0 0 0 00 0 F40 0 0 0 0 0 0 0 0                           IF SERIAL.PANEL *JUMPJ;   SERIAL.PANEL EXIT                     1618.000

                                                                PPATTN.RD.M1                                                            1619.000

0240 0 6 7 0 1 E E 0 02 0 306 0 0 0 0 0 0 0 0  306                      FULLMAR=(CMD.WDOT↑20&@FF000000),*LINK CMD.WDOT; CLEAR MAR        1620.000

                                                                *                       TRANSMIT PSW TO PARALLEL PANEL                  1621.000

0241 0 4 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  244                      *GO TO PP.END;                                                   1622.000

                                                                PP.NPRESENT                                                             1623.000

0242 0 0 0 7 1 E F 0 00 0 FF0 0 0 0 0 0 0 0 0                           T=-1;                                                           1624.000

0243 0 0 0 1 0 0 4 0 02 A 830 0 0 0 0 0 0 0 0                           SCRATCH(@83)=T;    SET NON-PRESENT PANEL INDICATOR              1625.000

                                                                PP.END                                                                  1626.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    P R O G R A M M E R   P A N E L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0244 9 4 0 0 1 0 0 0 00 D 5B6 0 0 0 0 0 0 0 0  246        OTHERBANK,IF %IONORESP:IOTIMEOUT *GO TO $+2;                    1627.000

0245 0 0 0 0 1 0 0 0 15 0 000 0 0 0 0 0 0 0 0             CLRTO;                                                         1628.000

0246 0 0 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0             RESET(HIREG);                                                  1629.000

0247 0 6 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0    4        *GO TO EVENT.POLL;  RETURN TO M AINSTREAM                       1630.000

                                                                                                                     1631.000
                                                  *                                                                  1632.000
                                                  *              PROGRAMMER PANEL WRITE FUNCTION                     1633.000
                                                  *                                                                  1634.000
                                                  *      CODES 0-7  EQUAL WRITE FILE REGISTERS 0-7                   1635.000
                                                  *      CODE  8    EQUALS WRITE PSD 1  (PSW 1)                      1636.000
                                                  *      CODES 9-A  N/A                                              1637.000
                                                  *      CODE  B    EQUALS WRITE PSD 2                               1638.000
                                                  *      CODE  C    EQUALS CONVERT ADDRESS TO PHYSICAL ADDRESS       1639.000

                                                                                                                     1640.000

                                                  PPATTN.WT                                                          1641.000

0248 0 0 0 3 1 E 1 0 02 0 020 0 0 0 0 0 0 0 0             S=@00000002;  SET DATA BUS 30 (SEND DATA)                       1642.000

0249 6 0 0 3 1 E B 1 02 0 000 0 0 0 0 0 0 0 0             FR(REV)=0;                                                     1643.000

024A 0 6 7 0 1 E F 0 00 0 33B 0 0 0 0 0 0 0 0  33B        T=S,*LINK RETRY.APSTX;                                         1644.000

024B 9 4 0 0 1 0 0 0 00 0 5BD 0 0 0 0 0 0 0 0  24D        IF %IONORESP:IOTIMEOUT *GO TO PP.PRESENT;                      1645.000

024C 0 4 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0  242        *GO TO PP.NPRESENT;                                            1646.000

                                                  PP.PRESENT                                                         1647.000

024D 0 0 0 3 4 D 8 0 02 0 0F0 0 0 0 0 0 0 0 0             R(TMP0)=@0000000F&R(TMP0);                                     1648.000

024E 0 0 0 7 0 3 0 0 12 0 F80 0 0 0 0 0 0 0 0             NOD=@FFFFFFF8 + T, SAVESIGN;    S/D CODE -8                    1649.000

024F 0 0 0 3 0 8 0 0 02 0 0C0 0 0 0 0 0 0 0 0             NOD=@0000000C ! T; TEST FOR CODE C (CONVERT ADDRESS)           1650.000

0250 B 4 0 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  258        IF ALU7 *GO TO PPATTN.WT.PSW;   ALU7 = WRITE PC                1651.000

0251 B 5 0 0 1 0 0 0 00 0 074 0 0 0 0 0 0 0 0  274        IF ALU7 *GO TO PPATTN.WT.CONV; GO TO CONVERT ADDRESS           1652.000

0252 3 4 0 0 1 0 0 0 00 1 006 0 0 0 0 0 0 0 0  256        IF SIGNSAVE *GO TO PPATTN.WT.REG; IF <8 GO TO FILE WRITE       1653.000

                                                  PP.WT.PSD2                                                         1654.000

0253 0 0 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0             RESET(HIREG);                                                  1655.000

0254 0 6 7 0 3 0 F 0 00 0 9CE 0 0 0 0 0 0 0 0  9CE        T=DI, *LINK PANEL.EVALUATE.PSD2; GO IMPLEMENT NEW PSD2         1656.000

0255 0 5 0 0 1 0 0 0 00 0 044 0 0 0 0 0 0 0 0  244        *GO TO PP.END;                                                 1657.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU PROGRAMMER PANEL

```
PC   T S M A B + D K  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1658.000
                                        *                                                                           1659.000
                                        *              WRITE FILE REGISTER SPECIFIED                                1660.000
                                        *                                                                           1661.000

                                                                                                                    1662.000
                                       PPATTN.WT.REG                                                                1663.000

0256 0 5 7 0 5 1 D 0 00 0 06E 0 0 0 0 0 0 0 0  26E      NL=%R(TMP0,HWS), *LINK DUD.2; ONES-COMP OF REG ADDR TO N.     1664.000

0257 0 5 0 0 3 0 B 3 00 0 044 0 0 0 0 0 0 0 0  244      R(NCTR)=DI, *GO TO PP.END;                                   1665.000

                                                                                                                    1666.000
                                        *                                                                           1667.000
                                        *          WRITE NEW PSW WORD 1 (USED BY PARALLEL AND SERIAL PANELS)        1668.000
                                        *                                                                           1669.000
                                        *      ENTRY PARAMETER :                                                    1670.000
                                        *        DI = NEW PSW 1 (FROM PARALLEL OR SERIAL PANELS)                    1671.000
                                        *                                                                           1672.000

                                                                                                                    1673.000
                                       PPATTN.WT.PSW                                                                1674.000

0258 0 0 0 4 3 D 2 0 00 D 800 0 0 0 0 0 0 0 0           PC=R(PCMASK)&DI, OTHERBANK;  LOAD ADDR FROM NEW PSW          1675.000

0259 6 0 0 0 1 E B 7 02 0 800 0 0 0 0 0 0 0 0           FR(TRACE)=@80000000;    RESET ALL FLAGS                      1676.000

025A 0 0 0 0 3 0 0 0 00 D 204 0 0 0 0 0 0 0 0           NOD=DI, RESET(PRIV), OTHERBANK; GO TO PRIV MODE              1677.000

025B F 3 0 3 3 D E 0 02 0 020 0 0 0 0 0 0 0 0           FULLMAK=@00000002&DI, IF FALSE *JUMPZ; TEST FOR PSW HALFWORD 1678.000

                                        *                       ;INDICATOR AND PURGE PIPELINE                       1679.000

025C 5 0 0 0 1 0 0 0 00 2 A04 0 0 0 0 0 0 0 0           IF %ALUNEGW SET(PRIV); GO UNPRIV IF NO PRIV BIT IN PSW1      1680.000

025D B 4 0 4 3 D 1 0 00 D 90F 0 0 0 0 0 0 0 0  25F      S=R(STMASK)&DI, OTHERBANK, IF ALUZ *GO TO $+2;GET CC'S & EXT BIT 1681.000

025E 6 0 0 2 4 A B 7 02 D 800 0 0 0 0 0 0 0 0           FR(TRACE)=@00008000:FR(TRACE), OTHERBANK; SET HALFWORD FLAG  1682.000

                                       SET.PSW.STATUS                                                               1683.000

025F 0 0 0 0 1 0 0 0 00 D 50C 0 0 0 0 0 0 0 0           OTHERBANK, SETXCC(S);                                        1684.000

0260 A 4 0 0 1 0 0 0 00 0 304 0 0 0 0 0 0 0 0  264      IF %MODE75 *GO TO WRT.RD.PSW.EXIT;                           1685.000

0261 0 0 0 0 3 D 0 0 02 0 010 0 0 0 0 0 0 0 0           NOD=@01000000&DI; TEST FOR ENABLE AEXP BIT IN PSW            1686.000

0262 0 0 0 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0           RESET(ENBL.AEXP);                                            1687.000

0263 C 0 0 0 1 0 0 0 00 0 B06 0 0 0 0 0 0 0 0           IF NALUZ SET(ENBL.AEXP);                                     1688.000

                                       WRT.RD.PSW.EXIT                                                              1689.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU PROGRAMMER PANEL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0264 1 4 0 0 1 0 0 0 00 0 F46 0 0 0 0 0 0 0 0  266        IF SERIAL.PANEL *HOP $+2;                                    1690.000

0265 0 5 0 0 1 0 0 0 00 0 044 0 0 0 0 0 0 0 0  244        *GO TO PP.END;   PARALLEL PANEL EXIT                         1691.000

0266 0 6 0 0 1 0 0 0 00 0 C23 0 0 0 0 0 0 0 0  C23        *GO TO S.P.READ.PSW;                                         1692.000

                                                                                                                      1693.000
                                                *                                                                     1694.000
                                                *              WHEN IPL COMPLETE THIS ROUTINE                         1695.000
                                                *              SENDS MEMORY ADDRESS OF CSW                            1696.000
                                                *              CELL TO PROGRAMMER PANEL                               1697.000
                                                *                                                                     1698.000

                                                                                                                      1699.000
                                                PP.CSW.ADDR1                                                          1700.000

0267 1 1 0 4 1 E F 0 02 6 F40 0 0 0 0 0 0 0 0              T=@F4FFFFFF, SDFST, IF SERIAL.PANEL *JUMPJ;                  1701.000

0268 9 4 0 0 0 7 F 0 00 F 58A 0 0 0 0 0 0 0 0  26A        T=S:%T(7E), IF %IONORESP:IOTIMEOUT *HOP $+2;                  1702.000

0269 0 4 0 0 1 0 0 0 15 0 00D 0 0 0 0 0 0 0 0  26D        CLRTO, *GO TO RESET.HIREG.RTN; CLEAR TIMEOUT AND EXIT        1703.000

026A 0 0 0 2 1 E F 0 02 0 070 0 0 0 0 0 0 0 0              T=@00000700;                                                 1704.000

026B 0 0 0 3 0 A F 0 02 0 800 0 0 0 0 0 0 0 0              T=@00000080:T; GENERATE CSWS ADDRESS                         1705.000

026C 0 6 7 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0  306        *LINK CMD.WOUT; GENERATE CSWS ADDRESS AND SEND TO PANEL      1706.000

                                                RESET.HIREG.RTN                                                       1707.000

026D 0 0 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0              RESET(HIREG);                                                1708.000

                                                DUD.2                                                                 1709.000

026E 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPJ;          DELAY REQUIRED ELSEWHERE                    1710.000

                                                PP.CSW.ADDR                                                           1711.000

026F 1 1 0 0 1 0 0 0 00 0 F40 0 0 0 0 0 0 0 0              IF SERIAL.PANEL *JUMPJ;                                      1712.000

0270 0 0 0 0 1 E E 0 02 0 804 0 0 0 0 0 0 0 0              FULLMAR=@80000000,SET(HIREG);                                1713.000

0271 6 0 0 3 1 E 8 1 02 0 000 0 0 0 0 0 0 0 0              FR(RDEV)=0; SET PANEL PHYSICAL ADDRESS                       1714.000

0272 0 0 0 3 1 E F 0 02 0 090 0 0 0 0 0 0 0 0              T=@00000009;              SET BITS 28 & 31 BUS DATA          1715.000

0273 0 6 0 0 0 0 0 0 00 0 337 0 0 0 0 0 0 0 0  337        *GO TO IO.SEQ, NOD=T;                                        1716.000

                                                                                                                      1717.000
                                                *                                                                     1718.000
                                                *  PARALLEL PANEL CONVERT ADDRESS FUNCTION                            1719.000
                                                *                                                                     1720.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU PROGRAMMER PANEL
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    1721.000
                                                    PPATTN.WT.CONV                                                  1722.000
0274 0 0 0 5 3 D E 0 02 0 0F0 0 0 0 0 0 0 0 0           FULLMAR=@FF0FFFFF&DT; GET THE LOGICAL ADDRESS                   1723.000
0275 0 0 0 0 0 0 C 0 00 F 805 0 0 0 0 0 0 0 0           NU=T(ZE), SET(ENAWORD); CLEAR N COUNTER                         1724.000
0276 0 5 7 0 1 0 3 0 00 0 06E 0 0 0 0 0 0 0 0  26E      MAR=MAR, *LINK DUP.2; MAP THE ADDRESS                           1725.000
0277 0 0 0 0 1 0 F 0 00 6 005 0 0 0 0 0 0 0 0           T=MAR, RDMAP, RESET(ENAWORD); GET THE CONTENTS OF PHYSICAL MAR  1726.000
0278 0 5 0 0 1 0 0 0 00 0 040 0 0 0 0 0 0 0 0  240      *GO TO PPATTN.RD.M1; GO TRANSMIT ADDRESS TO PARALLEL PANEL      1727.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MISCELLANEOUS INTERRUPTS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1728.000
                                                    CONSOLE.INTERRUPT                                               1729.000
0279 1 1 0 0 1 0 0 0 00 0 7E0 0 0 0 0 0 0 0 0           IF PROTV *JUMPJ;   GIVE PROTECT VIOL PRIORITY               1730.000
027A 0 0 0 1 1 E F 0 02 0 130 0 0 0 0 0 0 0 0           T=@00130000;     INTERRUPT LEVEL 13                         1731.000
027B 0 5 0 0 1 0 0 0 14 0 084 0 0 0 0 0 0 0 0  284      UPACK, *GO TO REQUEST.IT;                                   1732.000
                                                    ARITHMETIC.EXCEPTION                                            1733.000
027C 4 4 0 1 1 E F 0 02 C 29E 0 0 0 0 0 0 0 0  27E      T=@00290000, IF MODF75S *HOP CHECK.DISABLE.AEXP; INTERRUPT LVL29 1734.000
027D 0 5 0 0 1 0 0 0 16 0 084 0 0 0 0 0 0 0 0  284      RSTAEXP, *GO TO REQUEST.IT;                                 1735.000
                                                    CHECK.DISABLE.AEXP                                              1736.000
027E 0 0 0 3 1 E 6 0 02 0 400 0 0 0 0 0 0 0 0           DI=@00000040; SET DISABLE TRAP MASK                         1737.000
027F 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0           S=SCRATCH(@91); FETCH CPU STATUS WORD                       1738.000
0280 0 5 7 0 3 D 0 0 00 0 06F 0 0 0 0 0 0 0 0  26E      NOD=S&DI, *LINK DUD.2;  TEST FOR DISABLE AEXP                1739.000
0281 C 1 0 7 1 9 1 0 16 0 FF0 0 0 0 0 0 0 0 0           S=%@FFFFFFFF, RSTAEXP, IF NALUZ *JUMPJ; RETURN TO EXTERNAL EVENT 1740.000
                                                    *       PROCESSING IF AEXP IS DISABLED OR IF TRAPS ARE DISABLED  1741.000
0282 4 5 0 0 1 0 0 0 00 A 0AB 0 0 0 0 0 0 0 0  2AB      IF ENBL.AEXP *GO TO AEXP.STATUS.IPU;                        1742.000
0283 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPJ; RETURN IF AEXP IS NOT ENABLED.                      1743.000
                                                    REQUEST.IT                                                      1744.000
0284 0 0 0 0 1 0 0 0 00 0 804 0 0 0 0 0 0 0 0           SET(HIREG);                                                 1745.000
0285 0 0 0 0 0 0 8 0 00 0 A00 0 0 0 0 0 0 0 0           R(INTR)=T;                                                  1746.000
0286 0 6 7 0 1 0 0 0 00 0 467 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                                      1747.000
0287 6 0 0 2 4 A B 4 02 0 400 0 0 0 0 0 0 0 0           FR(OFFSET)=@00004000:FR(OFFSET); SET INTERRUPT RTN FLAG     1748.000
0288 0 6 0 0 1 0 0 0 00 0 44C 0 0 0 0 0 0 0 0  44C      *GO TO REQ.INTR;                                            1749.000
                                                    (@28A)                                                          1750.000
                                                    INSTR.SKIP                                                      1751.000
028A 6 0 0 6 4 B 0 7 02 0 FE0 0 0 0 0 0 0 0 0           NOD=%@FFFFFEFF&FR(TRACE); TEST INSTR SKIP BIT               1752.000
028B 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                       1753.000
028C B 1 0 0 1 0 0 0 00 8 000 0 0 0 0 0 0 0 0           PCTOMAR,IF ALUZ *JUMPJ;          NO RETURN                  1754.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MISCELLANEOUS INTERRUPTS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
028D 6 0 0 2 4 B B 7  02 0 810 0 0 0 0 0 0 0 0        FR(TRACE)=%@00008100&FR(TRACE);  TEST AND RESET RHFLAG          1755.000
028E 5 5 0 0 2 0 0 0  00 8 096 0 0 0 0 0 0 0 0 296    NOD=IO,IF %BMUX16 *GO TO NORHF;                               1756.000
028F 5 4 0 0 1 0 0 0  00 4 006 0 0 0 0 0 0 0 0 296    IF %BMUX00 *GO TO NORHF;                                      1757.000
0290 1 4 0 0 1 0 0 0  00 0 D86 0 0 0 0 0 0 0 0 296    IF FXFLAG *GO TO NORHF;                                       1758.000
0291 0 6 7 0 1 0 0 0  00 5 00F 0 0 0 0 0 0 0 0   F    TOGRHF,*LINK DUD.0XX;                                         1759.000
0292 A 4 0 0 1 0 0 0  00 0 A06 0 0 0 0 0 0 0 0 296    IF %RHFLAG *GO TO NORHF;                                      1760.000
0293 F 3 0 0 1 0 0 0  0C 0 000 0 0 0 0 0 0 0 0        RSTRHF,IF FALSE *JUMPZ;                                       1761.000
0294 6 0 0 2 4 A B 7  02 0 800 0 0 0 0 0 0 0 0        FR(TRACE)=@00008000:FR(TRACE);  SET RHFLAG IND                1762.000
0295 0 1 0 0 1 0 0 0  00 0 000 0 0 0 0 0 0 0 0        *JUMPJ;                                                       1763.000
                                                NORHF                                                           1764.000
0296 0 1 0 3 1 3 2 0  02 0 040 0 0 0 0 0 0 0 0        PC=@00000004+MAR,*JUMPJ;                                      1765.000
                                                IPL.CPU.MODE                                                    1766.000
0297 0 0 0 1 3 D F 0  02 0 550 0 0 0 0 0 0 0 0        T=@00550000&DI;                                               1767.000
0298 5 4 0 1 0 8 0 0  02 8 55B 0 0 0 0 0 0 0 0 29B    NOD=@00550000!T, IF %BMUX16 *HOP $+3;                         1768.000
0299 8 4 0 0 1 0 0 0  00 0 00F 0 0 0 0 0 0 0 0 29F    IF ALUZ *GO TO INITIALIZE.CPU.MODE;                           1769.000
029A C 4 0 0 1 0 0 0  00 0 00F 0 0 0 0 0 0 0 0 29F    IF NALUZ *GO TO INITIALIZE.CPU.MODE;                          1770.000
                                                IPL.55                                                          1771.000
029B 0 0 0 6 3 D F 0  02 F 7F0 0 0 0 0 0 0 0 0        T=@FFFF7FFF&DI(7E);                                           1772.000
029C 0 0 0 1 0 A F 0  02 0 550 0 0 0 0 0 0 0 0        T=@00550000:T;                                                1773.000
029D 0 0 0 1 0 0 4 0  02 A 840 0 0 0 0 0 0 0 0        SCRATCH(@84)=T;                                               1774.000
029E 0 0 0 0 0 0 8 0  00 0 900 0 0 0 0 0 0 0 0        R(RDEV)=T;                                                    1775.000
                                                INITTALIZE.CPU.MODE                                             1776.000
029F 1 4 0 0 1 0 0 0  00 0 F41 0 0 0 0 0 0 0 0 2A1    IF SERIAL.PANEL *HOP $+2;                                     1777.000
02A0 0 0 0 0 1 0 0 0  00 0 405 0 0 0 0 0 0 0 0        RESET(ENATBMT);                                               1778.000
02A1 0 0 0 1 1 0 1 0  02 A 840 0 0 0 0 0 0 0 0        S=SCRATCH(@84); FETCH DEFAULT IPL ENTRY                       1779.000
02A2 0 0 0 1 1 E F 0  02 0 FF0 0 0 0 0 0 0 0 0        T=@00FF0000;  LOAD FF TEST MASK                               1780.000
02A3 0 0 0 0 0 D F 0  00 0 004 0 0 0 0 0 0 0 0        T=S&T, RESET(HIREG);  TEST FOR  MODE                          1781.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
MISCELLANEOUS INTERRUPTS

```
SEL 32/75 CPU
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

02A4 4 4 0 0 1 0 0 0 00 2 009 0 0 0 0 0 0 0 0  2A9        IF IPU *GOTO INITIALIZE.75.EXIT;                                  1782.000
02A5 0 0 0 1 0 8 0 0 02 0 550 0 0 0 0 0 0 0 0              NOD=@00550000!T;                TEST FOR 55 MODE                  1783.000
02A6 0 0 0 3 1 E F 0 02 0 400 0 0 0 0 0 0 0 0              T=@00000040;  SET DISABLE TRAP CODE                               1784.000
02A7 B 4 0 1 0 0 4 0 02 A 91A 0 0 0 0 0 0 0 0  2AA        SCRATCH(@91)=T, IF ALU2 *HOP INITTALIZE.55.EXIT;                  1785.000
02A8 1 4 0 0 1 0 0 0 00 0 F3A 0 0 0 0 0 0 0 0  2AA        IF ENBL55 *HOP INITIALIZE.55.EXIT;                                1786.000
                                                       INITTIALIZE.75.EXIT                                               1787.000
02A9 0 1 0 0 1 0 0 0 00 0 806 0 0 0 0 0 0 0 0              SET(MODE75), *JUMPJ;   SET 75 MODE AND EXIT                       1788.000
                                                       INITTALIZE.55.EXIT                                                1789.000
02AA 0 1 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0              RESET(MODE75), *JUMPJ;                                            1790.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 79APR27
S F L 32/75 CPU R T O M - I N T E R V A L T I M E R
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    1791.000
                                                    AEXP.STATUS.IPU                                                 1792.000
                                                    *                                                               1793.000
02AB 4 4 0 0 1 0 0 0 00 2 00D 0 0 0 0 0 0 0 0  2AD          IF IPU *GOTO $+2;                                       1794.000
02AC 0 6 0 0 1 0 0 0 00 0 563 0 0 0 0 0 0 0 0  563          *GOTO QUEUE.TEST;          CPU MODE                     1795.000
02AD 0 0 0 2 1 E F 0 02 0 010 0 0 0 0 0 0 0 0               T=@00000100;               FETCH AEXP STATUS            1796.000
02AE 0 5 0 0 1 0 0 0 00 0 0D4 0 0 0 0 0 0 0 0  2D4          *GOTO IPU.ERROR.TRAP;                                   1797.000
                                                    (@2B0)                                                          1798.000
                                                    RTOM.RSTX                                                       1799.000
02B0 0 6 7 0 1 0 0 0 00 0 305 0 0 0 0 0 0 0 0  305        *LINK CMD.RSTX; LINK OUT AND EXECUTE RSTX TO RTOM             1800.000
02B1 0 6 7 0 1 0 0 0 00 0 303 0 0 0 0 0 0 0 0  303        *LINK TEST.BIBUSY;  GO WAIT FOR DRT OR TIMEOUT OR RETRY       1801.000
02B2 9 1 0 0 1 0 0 0 00 0 0B0 0 0 0 0 0 0 0 0              IF %IONORESP:IOCHBUSY:IOTIMEOUT:IORETRY *JUMPJ; EXIT IF XFER OK 1802.000
02B3 1 4 0 0 1 0 0 0 00 0 EB0 0 0 0 0 0 0 0 0  2B0        IF IORETRY *GO TO RTOM.RSTX; LOOP IF RETRY                    1803.000
                                                    RTOM.ERROR                                                      1804.000
02B4 0 6 0 0 1 0 0 0 00 0 3A4 0 0 0 0 0 0 0 0  3A4        *GOTO CD.EXIT;                                                1805.000
                                                    DEV.CLASS3                                                      1806.000
02B5 6 0 0 5 4 D D 3 02 0 000 0 0 0 0 0 0 0 0              NL=@FF00FFFF&FR(INTRTAB) ; TEST RAM LOADED & CLEAR N REG      1807.000
02B6 3 6 7 0 1 0 0 0 00 4 32E 0 0 0 0 0 0 0 0  32E        IF BMUX00 *LINK RAM.LOAD;                                     1808.000
02B7 A 4 0 0 1 0 0 0 00 0 904 0 0 0 0 0 0 0 0  2B4        IF %NCTRZ *GOTO RTOM.ERROR;  ERROR EXIT                       1809.000
02B8 6 0 0 3 4 A E 1 02 0 800 0 0 0 0 0 0 0 0              FULLMAR=@00000080:FR(RDEV); SET INTERVAL TIMER BIT IN DEST ADDR 1810.000
02B9 0 0 0 0 5 0 F 0 00 6 200 0 0 0 0 0 0 0 0              T=R(CMD,HWS), SDEST; FETCH CD INST BITS 16-31 IN T 0-15       1811.000
02BA 0 0 0 0 1 A E 0 00 F 000 0 0 0 0 0 0 0 0              FULLMAR=S:MAR(ZF); COMBINE CONTROL CODE & DEST ADDR           1812.000
02BB 0 4 7 3 1 E F 0 02 0 090 0 0 0 0 0 0 0 0  2B0        T=@00000009, *LINK RTOM.RSTX; FORMAT READ COUNT CODE          1813.000
02BC 0 0 0 0 5 0 F 0 00 6 200 0 0 0 0 0 0 0 0              T=R(CMD,HWS),SDEST; FETCH CD INST BITS 16-31                  1814.000
02BD 0 0 0 1 0 D 0 0 02 0 600 0 0 0 0 0 0 0 0              NOD=@00600000&T;  TEST FOR READ OR PROGRAM TIMER BITS         1815.000
02BE 0 0 0 1 0 D 0 0 02 0 400 0 0 0 0 0 0 0 0              NOD=@00400000&T;  TEST FOR READ TIMER BIT                    1816.000
02BF B 4 0 1 0 D 0 0 02 0 203 0 0 0 0 0 0 0 0  2C3        NOD=@00200000&T,IF ALUZ *HOP TIMER.EXIT;EXIT IF NOT RD OR PROG 1817.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
                                                                    R T O M - I N T E R V A L   T I M E R
S E L   3 2 / 7 5   C P U
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

02C0 B 4 0 0 4 0 1 0 00 D 002 0 0 0 0 0 0 0 0  2C2        S=R(0),OTHERBANK, IF ALUZ *HOP TIMER.CONTROL; FETCH COUNT IN R0   1818.000

02C1 B 4 0 0 3 0 8 0 00 0 003 0 0 0 0 0 0 0 0  2C3        R(0)=DI, IF ALUZ *HOP TIMER.EXIT; STORE READ TIMER COUNT IN R0   1819.000

                                                 TIMER.CONTROL                                                          1820.000

02C2 0 6 7 0 1 E F 0 00 D 34F 0 0 0 0 0 0 0 0  34F        T=S, OTHERBANK, *LINK IO.WDOT; LOAD TIMER COUNT IN T            1821.000

                                                 TIMER.EXIT                                                             1822.000

02C3 0 3 0 0 1 0 0 0 19 D 084 0 0 0 0 0 0 0 0              OTHERBANK, FETCHPC, RESET(HIREG), *JUMPZ;                     1823.000

                                                                                                                        1824.000
                                                 ***                                                                    1825.000
                                                 ***                75 MODE                                             1826.000
                                                 ***  FORMAT PROGRAM STATUS DOURLE WORD FOR BRI INSTRUCTION              1827.000
                                                 ***                                                                    1828.000
                                                 ***    ENTRY PARAMETER :                                               1829.000
                                                 ***      T = PSW 1                                                     1830.000
                                                 ***      R(PSD.ADDR) = REAL ADDRESS FOR PSD                            1831.000
                                                 ***                                                                    1832.000

                                                                                                                        1833.000

                                                 BRI.FORMAT.PSW                                                         1834.000

02C4 0 0 0 0 4 0 E 0 00 D 400 0 0 0 0 0 0 0 0              FULLMAR=R(PSD.ADDR), OTHERBANK; GET PSD ADDRESS                1835.000

02C5 0 0 0 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0              RESET(ENBL.AEXP);                                             1836.000

02C6 0 0 0 3 1 3 3 0 02 0 040 0 0 0 0 0 0 0 0              MAR=@0000004+MAR; VECTOR TO PSW 2 & ALLOW MAPPING             1837.000

02C7 0 0 0 4 0 D 2 0 1C 1 880 0 0 0 0 0 0 0 0              PC=R(PCMASK)&T, READ, FRCWORD; LOAD PC FROM PSW1 AND FETCH PSW2 1838.000

02C8 0 0 0 0 0 0 0 0 02 0 010 0 0 0 0 0 0 0 0              NOD=@01000000&T;  TEST FOR ENABLE AEXP BIT IN PSW 1           1839.000

02C9 0 0 0 0 6 0 8 0 00 0 C00 0 0 0 0 0 0 0 0              R(ALEVEL)=INTLVL; SAVE NEW ACTIVE LEVEL                       1840.000

02CA C 0 0 0 1 0 0 0 00 0 B06 0 0 0 0 0 0 0 0              IF NALUZ SET(ENBL.AEXP);                                      1841.000

02CB 0 6 7 0 3 0 F 0 00 0 9CF 0 0 0 0 0 0 0 0  9CF        T=DI, *LINK EVALUATE.PSD2; GO IMPLEMENT NEW PSD2               1842.000

                                                 BRI.75.EXIT                                                            1843.000

02CC 0 6 0 0 0 0 0 0 00 F 9C6 0 0 0 0 0 0 0 0  9C6        NOD=T(ZE), *GO TO LPSD2; CLEAR MAR AND EXIT THE BRI            1844.000
```

```
          S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                          R T O M - I N T E R V A L  T I M E R
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                          1845.000
                                        *                                                                 1846.000
                                        MAP.FAULT.TYPE                                                    1847.000
                                        *                                                                 1848.000
02CD 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0                T=@00000000;                 PREPARE IPU STATUS FLAG 1849.000
02CE 4 5 0 0 1 0 0 0 00 2 0D4 0 0 0 0 0 0 0 0  2D4           IF IPU *GOTO IPU.ERROR.TRAP;                      1850.000
02CF 0 6 0 0 1 0 0 0 00 0 580 0 0 0 0 0 0 0 0  580           *GOTO TRAP.MODE75;                                1851.000
                                        *                                                                 1852.000
                                        *                                                                 1853.000
                                        PRIV.TYPE                                                         1854.000
                                        *                                                                 1855.000
02D0 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0                RESET(ENAUORD);                                  1856.000
02D1 4 4 0 0 1 0 0 0 00 2 003 0 0 0 0 0 0 0 0  2D3           IF IPU *GOTO $+2;                                 1857.000
02D2 0 6 0 0 1 0 0 0 00 0 563 0 0 0 0 0 0 0 0  563           *GOTO QUEUE.TEST;          CONTINUE AS CPU        1858.000
02D3 0 0 0 2 1 E F 0 02 0 040 0 0 0 0 0 0 0 0                T=@00000400;                FETCH PRIV VIOL STATUS 1859.000
                                        *                                                                 1860.000
                                        IPU.ERROR.TRAP                                                    1861.000
                                        *                                                                 1862.000
02D4 0 0 0 3 0 A F 0 02 0 100 0 0 0 0 0 0 0 0                T=@00000010:T;               INSERT IPU STATUS  1863.000
02D5 0 6 7 0 1 0 0 0 00 0 596 0 0 0 0 0 0 0 0  596           *LINK UPDATE.CPU.STATUS;                          1864.000
02D6 0 0 0 3 1 E F 0 02 0 EC0 0 0 0 0 0 0 0 0                T=@000000EC;                                      1865.000
02D7 0 0 0 2 0 A F 0 02 6 020 0 0 0 0 0 0 0 0                T=@00000200:T,SDEST;        FETCH ERROR TRAP      1866.000
02D8 0 6 0 0 1 0 0 0 00 0 580 0 0 0 0 0 0 0 0  580           *GOTO TRAP.MODE75;                                1867.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU I/O REGISTER LOAD

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1868.000
                                                **                                                                  1870.000
                                                ** LOAD I/O REGISTERS                                               1871.000
                                                **     ENTRY PARAMETERS :                                           1872.000
                                                **       T = CHANNEL OR DEVICE ADDR IN BITS 9-15                    1873.000
                                                **                                                                  1874.000
                                                **     EXIT PARAMETERS :                                            1875.000
                                                **       R(EDEV) = CHANNEL OR DEVICE ADDR IN BITS 9-15              1876.000
                                                **       R(RDEV) = SCRATCH PAD DEVICE ENTRY                         1877.000
                                                **       R(INTR) = CHANNEL OR DEVICE INTERRUPT LEVEL IN BITS 9-15   1878.000
                                                **       R(INTRLOC) = SCRATCH PAD INTERRUPT ENTRY ADDR              1879.000
                                                **       R(INTRTAB) = SCRATCH PAD INTERRUPT ENTRY                   1880.000
                                                **       R(TI) = TCW (TI) MEMORY ADDR (NON-CLASS F ONLY)            1881.000
                                                **       R(IOCD.ADDR) = IOCD MEMORY ADDRESS                         1882.000
                                                **       R(TCW) = TCW  (CONTENTS OF TI LOCATION)  (NON-CLASS F ONLY) 1883.000
                                                **       R(SI.VECTOR) = SI VECTOR (CONTENTS OF VECTOR ADDR)         1884.000
                                                **                      ( 75 MODE ONLY)                             1885.000
                                                **                                                                  1886.000

                                                                                                                    1887.000
                                                (@2DB)                                                              1888.000
                                                IO.REG.LOAD                                                         1889.000
02DB 0 0 0 0 0 0 8 0 00 6 800 0 0 0 0 0 0 0 0          R(EDEV)=T,SDEST;        SAVE DEVICE ADDRESS SET SCR ADDR    1890.000
02DC 9 4 0 0 1 0 1 0 00 A BDE 0 0 0 0 0 0 0 0  2DE     S=SCRATCH(S), IF %BIBUSY *HOP $+2; VECTOR TO DEVICE TABLE   1891.000
02DD 1 4 0 0 1 0 0 0 00 0 BDD 0 0 0 0 0 0 0 0  2DD     IF BIBUSY *HOP S;                                           1892.000
02DE 0 0 0 0 1 E 8 0 00 0 900 0 0 0 0 0 0 0 0          R(RDEV)=S;              STORE DEVICE ENTRY                  1893.000
02DF 6 0 0 1 0 C B 2 02 0 7F0 0 0 0 0 0 0 0 0          FR(INTR)=@007F0000&%T;       ISOLATE INTERRUPT LEVEL        1894.000
02E0 0 0 0 0 1 E 6 0 00 0 000 0 0 0 0 0 0 0 0          DI=S;   MOVE DVC ENTRY AND TEST FOR ZERO                    1895.000
02E1 6 0 0 1 0 A B 5 02 6 800 0 0 0 0 0 0 0 0          FR(INTRLOC)=@00800000:T, SDEST; MAKE SCRATCH INT ENTRY ADDR 1896.000
02E2 B 6 0 0 1 0 1 0 00 A 485 0 0 0 0 0 0 0 0  485     S=SCRATCH(S), IF ALU2 *GO TO INVALID.IOM; FETCH INT TABLE ENTRY 1897.000
02E3 0 0 0 0 1 E 8 0 00 0 800 0 0 0 0 0 0 0 0          R(INTRTAB)=S;  SAVE INTERRUPT ENTRY                         1898.000
02E4 0 0 0 0 0 0 E 0 00 F 00D 0 0 0 0 0 0 0 0          FULLMAR=T(ZE), CLDNU;  LOAD SI VECTOR ADDR IN MAR           1899.000
02E5 0 0 0 0 3 1 0 0 1C 1 080 0 0 0 0 0 0 0 0          NOU=%DI, READ, FRCWORD; FETCH SI VECTOR AND TEST FOR CLASS F I/O 1900.000
02E6 6 0 0 6 0 0 B 4 02 0 000 0 0 0 0 0 0 0 0          FR(OFFSET)=@FFFF00FF&T; CLEAN BITS 16-23 FROM INT ENTRY     1901.000
02E7 A 4 0 0 1 0 0 0 00 0 70E 0 0 0 0 0 0 0 0  2EE     IF %ALU4-7Z *GO TO NON.EXT.IO.REG.LOAD; EXIT IF NOT CLASS F 1902.000
                                                EXT.IO.REG.LOAD                                                     1903.000
02E8 0 0 0 4 3 D F 0 00 D 800 0 0 0 0 0 0 0 0          T=R(PCMASK)&DI, OTHERBANK; CLEAN SI VECTOR TO VALID ADDRESS 1904.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SFL  32/75  CPU                                          I/O  REGISTER  LOAD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

02E9 0 0 0 3 0 3 E 0 02 0 100 0 0 0 0 0 0 0 0              FULLMAR=@00000010+T; COMPUTE  ADDR OF IOCD.ADDR                1905.000

02EA 0 4 7 0 0 0 8 0 1C 1 68C 0 0 0 0 0 0 0 0  2EC        R(SI.VECTOR)=T, READ, FRCWORD, *LINK IO.REG.LOAD.EXIT; FETCH      1906.000

                                                *                IOCD ADDRESS AND SAVE SI VECTOR                   1907.000

02EB 0 5 7 0 3 0 8 0 00 0 06F 0 0 0 0 0 0 0 0  26E        R(IOCD.ADDR)=DI, *LINK DUD.2;   SAVE IOCD ADDR              1908.000

                                               IO.REG.LOAD.EXIT                                                   1909.000

02EC 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0              IF %OPNORESP:OPTIMEOUT:OPRNOPE *JUMPJ; TEST FOR MEMORY ERRORS     1910.000

02ED 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530        NU=T(ZE), *GO TO CURRENT.INST.ERROR;                      1911.000

                                                                                                                  1912.000
                                                **                                                                1913.000
                                                **  LOCATE I/O DEDICATED MEMORY ADDRESS  (NON-CLASS F ONLY)       1914.000
                                                **                                                                1915.000
                                                **     ENTRY PARAMETERS :                                         1916.000
                                                **       R(INTR) = CHANNEL INTERRUPT LEVEL IN BITS 9-15           1917.000
                                                **                                                                1918.000
                                                **     EXIT PARAMETERS :                                          1919.000
                                                **       R(TI) = TCW MEMORY ADDRESS                               1920.000
                                                **       R(SI) = SI MEMORY ADDRESS                                1921.000
                                                **       R(TCW) = ACTUAL TCW (CONTENTS OF TI LOCATION)            1922.000
                                                **       R(IOCD.ADDR) = IOCD MEMORY ADDRESS OR STATUS LOCATIONS FOR 1923.000
                                                **                      EXTENDED I/O.                             1924.000
                                                **                                                                1925.000

                                                                                                                  1926.000

                                               NON.EXT.TO.REG.LOAD                                                1927.000

02EE 6 0 0 7 5 3 1 2 00 F EC0 0 0 0 0 0 0 0 0              S=@FFFFFFEC+FR(INTR,HWS,7E); ADD -@14 TO INTERRUPT LEVEL          1928.000

02EF 0 0 0 7 1 3 F 0 00 0 C00 0 0 0 0 0 0 0 0              T=@FFFFFFC0+MAR ; ADD -40 TO SI.IVL                               1929.000

02F0 0 0 0 1 3 E 1 0 00 0 000 0 0 0 0 0 0 0 0              S=SLEFT, BMUX=DI; *2 TO I/O INTERRUPT POINTER & WAIT FOR MEM      1930.000

02F1 0 0 0 0 0 0 8 0 01 0 600 0 0 0 0 0 0 0 0              R(TI)=T, SHIFTS(SLL); STORE TI ADDR , *4 TO POINTER               1931.000

02F2 0 0 0 0 0 0 E 0 00 0 000 0 0 0 0 0 0 0 0              FULLMAR=T;                                                        1932.000

02F3 0 4 7 0 1 0 0 0 1C 1 088 0 0 0 0 0 0 0 0  2F8        READ,FRCWORD,*LINK N.EXT.IO.REG.LOAD.EXIT; FETCH TCW FROM MEMORY  1933.000

02F4 0 0 0 1 1 E F 0 00 0 000 0 0 0 0 0 0 0 0              T=SLEFT; MAKE DOUBLE WORD POINTER (*8 TOTAL TO POINTER)           1934.000

02F5 0 0 0 1 1 0 1 0 02 A 820 0 0 0 0 0 0 0 0              S=SCRATCH(@82);  FETCH IOCD BASE ADDR                             1935.000

02F6 0 0 0 0 0 3 8 0 00 0 000 0 0 0 0 0 0 0 0              R(IOCD.ADDR)=S+T; STORE IOCD ADDR (BASE ADDR + POINTER)           1936.000

02F7 0 5 7 0 3 0 8 0 00 0 36E 0 0 0 0 0 0 0 0  26E        R(TCW)=DI, *LINK DUD.2;                                           1937.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU I/O REGISTER LOAD

```
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR

                                                N.EXT.IO.REG.LOAD.EXIT                                                  1938.000

02F8 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0         IF %OPNORESP:OPTIMEOUT:OPRNDPE *JUMPJ; TEST FOR MEMORY ERRORS     1939.000

02F9 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530    NU=T(ZE), *GO TO CURRENT.INST.ERROR;                              1940.000

                                                                                                                        1941.000
                                                *                                                                       1942.000
                                                *   CLEAR MEMORY SAVES ANY PREVIOUS F-C BITS AND FORCES A READ          1943.000
                                                *    TO MEMORY MODULE ZERO TO CLEAR ANY EXISTING ERRORS.  THE           1944.000
                                                *    READ USES THE PREVIOUS F-C BITS.  IF MODULE ZERO IS MAPPED         1945.000
                                                *    TO NON-PRESENT MEMORY, ADDITIONAL ERRORS WILL OCCUR.               1946.000
                                                *                                                                       1947.000

                                                                                                                        1948.000

                                                CLEAR.MEM.ERROR                                                         1949.000

02FA 1 4 0 0 1 0 0 0 00 0 BDA 0 0 0 0 0 0 0 0  2FA    IF BIBUSY *GO TO S;                                               1950.000

02FB 5 4 0 3 1 0 6 0 02 F 03D 0 0 0 0 0 0 0 0  2FD    DI=@00000003%MAP, IF %BYTE *HOP $+2; GET MAR C BITS               1951.000

02FC 0 0 0 1 3 A 6 0 02 0 080 0 0 0 0 0 0 0 0         DI=@00080000:DI;  FORCE A F BIT                                   1952.000

02FD 0 0 0 0 3 0 3 0 00 0 805 0 0 0 0 0 0 0 0         MAR=DI, SET(ENAIOPD);              LOAD F AND C BITS              1953.000

02FE 0 0 0 0 1 0 0 0 1C 4 085 0 0 0 0 0 0 0 0         READ, RSTPROTV, RESET(ENAIORD);  CLEAR MAP INVALID                1954.000

02FF 0 5 0 0 1 0 0 0 00 0 022 0 0 0 0 0 0 0 0  322    *GO TO DUD.WAIT.DRT;                                              1955.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU COMMON BUS TRANSFERS

```
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    1956.000

                                                                                                                    1957.000
                                                    **                                                              1958.000
                                                    **          MEMORY WRITE ROUTINE                                1959.000
                                                    **                                                              1960.000

                                                                                                                    1961.000

                                                    (@300)                                                          1962.000

                                                    MEMORY.WRITE                                                    1963.000

0300 0 4 0 0 1 0 0 0 1E 0 082 0 0 0 0 0 0 0 0  302      WRITE, *HOP DELAYED.BI.BUSY.TEST;                          1964.000

                                                                                                                    1965.000
                                                    **                                                              1966.000
                                                    **            **ISSUE ARTSX**                                   1967.000
                                                    **                                                              1968.000

                                                                                                                    1969.000

                                                    CMD.ARSTX                                                       1970.000

0301 0 0 0 0 1 0 0 0 1E 0 040 0 0 0 0 0 0 0 0            ARSTX;                                                     1971.000

                                                    DELAYED.BI.BUSY.TEST                                            1972.000

0302 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0            *NOP;                                                      1973.000

                                                    TEST.BIBUSY                                                     1974.000

0303 1 4 0 0 1 0 0 0 00 0 BD3 0 0 0 0 0 0 0 0  303       IF BIBUSY *GO TO $;                                        1975.000

                                                    CD.DUD                                                          1976.000

0304 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0            *JUMPJ;                                                    1977.000

                                                                                                                    1978.000
                                                    **                                                              1979.000
                                                    **            **ISSUE RSTX**                                    1980.000
                                                    **                                                              1981.000

                                                                                                                    1982.000

                                                    CMD.RSTX                                                        1983.000

0305 0 4 0 0 1 0 0 0 1C 0 049 0 0 0 0 0 0 0 0  309      RSTX, *HOP TRIPLE.DUD;                                     1984.000

                                                                                                                    1985.000
                                                    **                                                              1986.000
                                                    **            **ISSUE WDOT**                                    1987.000
                                                    **                                                              1988.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU COMMOM BUS TRANSFERS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                1989.000

                                                                                1990.000
                                        CMD.WDOT
                                                                                1991.000
0306 0 4 0 0 1 0 0 0 1E 0 002 0 0 0 0 0 0 0 0  302      WDOT, *HOP DELAYED.BI.BUSY.TEST;

                                                                                1992.000
                                                                                1993.000
                                        **                                      1994.000
                                        **          **ISSUE AICT**              1995.000
                                        **

                                                                                1996.000

                                        CMD.AICT                                1997.000

0307 0 4 0 0 1 0 0 0 1F 0 012 0 0 0 0 0 0 0 0  302      AICT, *HOP DELAYED.BI.BUSY.TEST;  1998.000

                                                                                1999.000
                                                                                2000.000
                                        **                                      2001.000
                                        **          **ISSUE ICT**               2002.000
                                        **

                                                                                2003.000

                                        CMD.ICT                                 2004.000

0308 0 4 0 0 1 0 0 0 1C 0 019 0 0 0 0 0 0 0 0  309      ICT, *HOP TRIPLE.DUD;     2005.000

                                        TRIPLE.DUD                              2006.000

0309 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP  ;  DELAY 3 CLOCKS   2007.000

                                        DOUBLE.DUD                              2008.000

030A 0 4 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  304      *GO TO CD.DUD;            2009.000

                                                                                2010.000
                                        **                                      2011.000
                                        **          FULLWORD MEMORY READ        2012.000
                                        **

                                                                                2013.000

                                        MEMORY.READ                             2014.000

030B 0 4 0 0 1 0 0 0 1C 1 082 0 0 0 0 0 0 0 0  302      READ, FRCWORD, *HOP DELAYED.BI.BUSY.TEST;  2015.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                    N O N - C L A S S   F   I / O
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    2016.000

                                                                                                                    2017.000
                                                    **                                                              2018.000
                                                    **              TEST RETRY                                      2019.000
                                                    **                                                              2020.000

                                                                                                                    2021.000

                                                    TEST.RETRY                                                      2022.000

030C 1 4 0 0 1 0 0 0 00 0 0AF 0 0 0 0 0 0 0 0  30F          IF FFINT *GO TO TEST.IPL;                                  2023.000

030D 2 1 0 0 1 0 0 0 00 0 B00 0 0 0 0 0 0 0 0                IF NCTR4 *JUMPJ;                                           2024.000

030E 0 4 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  304           *GO TO CD.DUD;  (PROVIDE TWO EXTRA CLOCKS IN THE RETRY LOOP  2025.000

                                                    TEST.IPL                                                        2026.000

030F 0 0 0 0 5 0 0 0 00 D F00 0 0 0 0 0 0 0 0                MOD=R(TRACE,HWS),OTHERBANK;     TEST IPL BIT               2027.000

0310 3 5 0 0 1 E E 0 02 B 356 0 0 0 0 0 0 0 0  356           FULLMAR=(CALM.EXTL!20R@FF000000),IF BMUX19 *GO TO CALM.EXTL;  2028.000

0311 0 5 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  30D           *GO TO TEST.RETRY+1;                                       2029.000

                                                                                                                    2030.000
                                                    **                                                              2031.000
                                                    **          86 TCW FORMATER                                     2032.000
                                                    **            FOR CLASS 0, 1, AND 2 (TLC) DEVICES               2033.000
                                                    **             ENTRY PARAMETER :                                2034.000
                                                    **               R(TCW) = TCW                                   2035.000
                                                    **                                                              2036.000

                                                                                                                    2037.000

                                                    FORMAT.TCW                                                      2038.000

0312 6 0 0 3 4 A 1 0 02 D 030 0 0 0 0 0 0 0 0                S=@00000003:FR(PCMASK), OTHERBANK; SET S = @0007FFFF       2039.000

0313 0 0 0 0 4 0 3 0 00 0 300 0 0 0 0 0 0 0 0                MAR=R(TCW); LOAD MAR FOR F-C BIT TEST                      2040.000

0314 0 0 0 0 4 D F 0 00 0 300 0 0 0 0 0 0 0 0                T=SAR(TCW); GET CLEAN DATA ADDRESS                         2041.000

0315 0 0 0 0 5 0 1 0 1C F 380 0 0 0 0 0 0 0 0                S=R(TCW,HWS,ZE), READ; GET XFER COUNT IN BITS 16-17,TEST F-C  2042.000

0316 0 0 0 0 0 0 8 0 00 0 100 0 0 0 0 0 0 0 0                R(D.ADDR)=T; SAVE DATA ADDRESS                             2043.000

                                                    **              CONVERT TRANSFER COUNT TO BYTE COUNT             2044.000

0317 3 4 0 3 3 E 1 0 00 F 00B 0 0 0 0 0 0 0 0  318           S=SNIBR, BMUX=DI, IF BYTE *GO TO $+4; XFER COUNT TO BITS 20-31  2045.000

0318 3 4 0 0 1 0 0 0 00 E 00A 0 0 0 0 0 0 0 0  31A           IF HWORD *GO TO $+2;                                       2046.000

0319 0 0 0 0 1 0 0 0 01 0 010 0 0 0 0 0 0 0 0                SHIFTS(SLC);                                               2047.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U N O N - C L A S S F I / O

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

031A 0 0 0 0 1 0 0 0 01 0 010 0 0 0 0 0 0 0 0          SHIFTS(SLC);                                                    2048.000
031B 1 4 0 0 1 E 0 0 00 0 F6D 0 0 0 0 0 0 0 0  31D     NOD=S, IF MAPINVALID *HOP $+2; TEST FOR XFER COUNT =0           2049.000
031C 9 4 0 0 1 0 0 0 00 0 55F 0 0 0 0 0 0 0 0  31F     IF %OPNOKFSP:OPTIMEOUT:OPRNOPE *HOP $+3;                        2050.000
031D 0 6 7 0 1 0 0 0 00 0 2FA 0 0 0 0 0 0 0 0  2FA     *LINK CLEAR.MFM.ERROR;                                          2051.000
031E 0 4 0 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  31B     *GO TO $-3;                                                     2052.000
031F C 4 0 2 1 E F 0 02 0 101 0 0 0 0 0 0 0 0  321     T=@00001000, IF NALUZ *GO TO $+2;                               2053.000
0320 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0          S=T;  SET MAX XFER COUNT IF INITIAL =0                          2054.000
0321 0 0 0 0 1 E 8 0 00 0 300 0 0 0 0 0 0 0 0          R(TCW)=S ; SAVE TRANSFER COUNT                                  2055.000
                                                DUD.32X                                                            2056.000
                                                DUD.WAIT.DRT                                                       2057.000
0322 0 1 0 0 3 0 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=DI,*JUMPJ;                                                  2058.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MON-CLASS F I/O

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2059.000
                                                    **                                                              2060.000
                                                    **      ENTRY PARAMETER :                                       2061.000
                                                    **        T = CALLER'S REQUEST                                  2062.000
                                                    **                                                              2063.000
                                                    CMD.IOC                                                         2064.000
0323 0 0 0 0 4 0 0 0 00 0 800 0 0 0 0 0 0 0 0                       NOD=R(INTRTAB); TEST FOR RAM LOADED                 2065.000
0324 5 5 0 0 0 0 8 0 00 4 737 0 0 0 0 0 0 0 0  337                  R(BUSREQ)=T, IF %BMUX00 *GO TO IO.SFQ; SAVE CALLER'S REQUEST 2066.000
0325 0 4 7 0 0 0 C 0 00 F 00E 0 0 0 0 0 0 0 0  32E                  NU=T(ZE),*LINK RAM.LOAD;                            2067.000
0326 3 5 0 0 1 E E 0 02 0 356 0 0 0 0 0 0 0 0  356                  FULLMAR=(CALM.EXTL↑20&@FF000000), IF EXTLW *GO TO CALM.EXTL; 2068.000
0327 A 1 0 0 1 0 0 0 00 0 900 0 0 0 0 0 0 0 0                       IF %NCTRZ *JUMPJ; ERROR EXIT                        2069.000
0328 0 5 0 0 4 0 F 0 00 0 737 0 0 0 0 0 0 0 0  337                  T=R(BUSREQ),*GO TO IO.SFW; RESTORE CALLERS REQUEST   2070.000

                                                                                                                    2071.000
                                                    ***                                                             2072.000
                                                    ***     MAKE SURE READY FROM PREVIOUS REQUEST NOT PENDING        2073.000
                                                    ***                                                             2074.000
                                                    COUNT.DOWN                                                      2075.000
0329 6 0 0 1 4 D 0 3 02 0 800 0 0 0 0 0 0 0 0                       NOD=@00800000&ER(INTRTAB); TEST FOR RTOM ENTRY      2076.000
032A 1 1 0 0 0 0 0 0 00 F 7AD 0 0 0 0 0 0 0 0                       NOD=1(ZE), IF IORFSPRDY *JUMPJ, CLDNU; EXIT NOT READY PENDING 2077.000
032B 1 4 0 0 1 0 0 0 00 0 7A9 0 0 0 0 0 0 0 0  329                  IF IORESPRDY *HOP COUNT.DOWN; TEST FOR READY RECEIVED 2078.000
032C A 4 3 0 1 0 0 0 00 0 90B 0 0 0 0 0 0 0 0  32B                  DECRN, IF %NCTR7 *GO TO $-1;  WAIT FOR READY        2079.000
032D 0 4 0 0 1 0 0 0 15 0 009 0 0 0 0 0 0 0 0  329                  CLRTO, *GO TO COUNT.DOWN;                           2080.000
                                                                                                                    2081.000
                                                    **                                                              2082.000
                                                    **      GENERATE RAM LOAD SEQUENCE                              2083.000
                                                    **                                                              2084.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          N O N - C L A S S  F  I / O
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    RAM.LOAD                                                                    2085.000

032E 6 0 0 2 4 D 1 1 02 0 FF0 0 0 0 0 0 0 0 0                S=@0000FF00&FR(RDFV);              CLEAR ALL BUT IOC ADDR      2086.000

032F 0 0 0 3 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0                S=SNIBR; SHIFT PHYSICAL ADDR                                   2087.000

0330 6 0 0 0 4 A B 3 02 0 800 0 0 0 0 0 0 0 0                FR(INTRTAB)=@80000000:FR(INTRTAB);  SET RAM LOADED BIT         2088.000

0331 B 4 0 3 1 E F 0 00 0 003 0 0 0 0 0 0 0 0  333           T=SNIBR, IF ALU7 *GO TO RAM.LOAD.EXIT; SHIFT PHYSICAL ADDR     2089.000

0332 0 4 0 0 0 A F 0 02 0 407 0 0 0 0 0 0 0 0  337           T=@40000000:1,*GO TO IO.SEQ; FORMAT WDOT LOAD RAM              2090.000

                                                    RAM.LOAD.EXIT                                                               2091.000

0333 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304           NU=%I(7E), *GO TO CO.DUD; ERROR & SCPI EXIT                    2092.000
```

```
          S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    N O N - C L A S S  F  I / O
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                        2093.000
                                                **                                                      2094.000
                                                **      CHECK FOR IPU DURING POWER EXIT                 2095.000
                                                **                                                      2096.000
                                                (@334)                                                  2097.000
                                                IPU.PWR.OUT                                             2098.000
0334 4 6 0 0 1 0 0 0 00 2 08F 0 0 0 0 0 0 0 0   BF          IF IPU *GO TO STOP; PREPARE IPU TO HALT     2099.000
0335 6 0 0 2 4 A 6 7 02 0 100 0 0 0 0 0 0 0 0               DI=@00001000:FR(TRACE);  SET POWER FAILSAFE BIT 2100.000
0336 0 6 0 0 1 0 0 0 00 0 111 0 0 0 0 0 0 0 0  111         *GO TO PWR.OFF.OUT+1; RETURN TO POWER OFF EXIT  2101.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU NON-CLASS F I/O
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    2102.000

                                                                                                                    2103.000
                                                **                                                                  2104.000
                                                **      I/O SEQUENCE    **                                          2105.000
                                                **        ENTRY PARAMETER :                                         2106.000
                                                **            T = CALLERS REQUEST                                   2107.000
                                                **                IF T CONTAINS BITS IN 27-31, AN ARSTX & RSTX SEQUENCE 2108.000
                                                **                   IS EXECUTED.                                   2109.000
                                                **                                                                  2110.000
                                                **                IF T CONTAINS BITS IN 00-03, AN ARSTX, RSTX, & WDOT 2111.000
                                                **                   SEQUENCE IS EXUWD.                             2112.000
                                                **                                                                  2113.000
                                                                                                                    2114.000
                                                (@337)                                                              2115.000
                                                IO.SEQ                                                              2116.000
0337 0 0 0 0 0 D 0 0 02 0 F00 0 0 0 0 0 0 0 0           NOD=@F0000000RT;        TEST IF WDOT REQUEST                2117.000
0338 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0           S=T;                                                        2118.000
                                                **              IF WDOT REQUEST INSERT A REQUEST FOR SUB-ADDR       2119.000
                                                **              AND CONTROLLER STATUS FOR THE ARSTX/RSTX SEQUENCE   2120.000
0339 C 4 0 3 1 E F 0 02 0 08B 0 0 0 0 0 0 0 0 336       T=@00000008, IF NALUZ *GO TO RETRY.ARSTX;                   2121.000
033A 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0           T=S;            RESTORE ORIGINAL REQUEST                    2122.000
                                                RETRY.ARSTX                                                         2123.000
033B 0 5 7 0 0 0 C 0 00 F 029 0 0 0 0 0 0 0 0 329       NU=T(ZE), *LINK COUNT.DOWN;                                 2124.000
033C 0 5 7 0 4 0 E 0 00 0 901 0 0 0 0 0 0 0 0 301       FULLMAR=R(RDEV),*LINK CMD.ARSTX;                            2125.000
033D 9 4 0 0 1 0 0 0 00 0 1BF 0 0 0 0 0 0 0 0 33F       IF %IONORESP:IOTIMEOUT:IOCHBUSY *HOP $+2;                   2126.000
033E 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0 304       NU=%T(ZE), *GO TO CD.DUD; ERROR OR BUSY EXIT                2127.000
033F 9 4 0 0 1 0 0 0 00 0 EB2 0 0 0 0 0 0 0 0 342       IF %IORETRY *GO TO $+3;                                     2128.000
0340 0 5 7 0 1 0 0 0 08 0 00C 0 0 0 0 0 0 0 0 30C       INCRN,*LINK TEST.RETRY;                                     2129.000
0341 0 5 0 0 1 0 0 0 00 0 03C 0 0 0 0 0 0 0 0 33C       *GO TO RETRY.ARSTX+1;                                       2130.000
                                                WAIT.RDY                                                            2131.000
0342 9 4 0 0 1 0 0 0 00 0 5A2 0 0 0 0 0 0 0 0 342       IF %IORESPRDY:FFINT *GO TO $;                               2132.000
0343 1 4 0 0 1 0 0 0 00 0 7A5 0 0 0 0 0 0 0 0 345       IF IORESPRDY *GO TO RDY.HERE;                               2133.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU NON-CLASS F I/O

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0344 0 5 0 0 1 0 0 0 00 0 056 0 0 0 0 0 0 0 0  356        *GO TO IORESPRDY.PENDING;                                          2134.000

                                                      RDY.HERE                                                           2135.000

0345 0 5 7 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0  305        *LINK CMD.RSTX;                                                    2136.000

0346 9 4 0 0 1 0 0 0 00 0 E69 0 0 0 0 0 0 0 0  349        IF %IORETRY *GO TO $+3;                                            2137.000

0347 0 5 7 0 1 0 0 0 0B 0 00C 0 0 0 0 0 0 0 0  30C        INCRN,*LINK TEST.RETRY;                                            2138.000

0348 0 4 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0  345        *GO TO RDY.HERE;                                                   2139.000

0349 1 4 0 0 1 E F 0 00 0 BD9 0 0 0 0 0 0 0 0  349        T=S, IF BIBUSY *HOP $;  WAIT FOR DRT                               2140.000

034A 9 4 0 0 3 0 0 0 00 E 1BC 0 0 0 0 0 0 0 0  34C        NOD=DI(SE), IF %IONORESP:IOTIMEOUT:IOCHBUSY *HOP $+2;               2141.000

034B 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304        NU=%T(ZE), *GO TO CD.DUD; ERROR OR BUSY EXIT                       2142.000

034C 0 0 0 0 0 0 0 0 02 0 F00 0 0 0 0 0 0 0 0              NOD=@F0000000&T; TEST FOR WDOT REQUEST                             2143.000

034D 2 5 0 0 3 0 C 0 00 F 822 0 0 0 0 0 0 0 0  322        NU=DI(ZE), IF ALUNEG *GO TO DUD.WAIT.DRT;  DEVICE BUSY--RETURN     2144.000

034E B 1 0 0 3 0 0 0 00 0 000 0 0 0 0 0 0 0 0              NOD=DI, IF ALUZ *JUMP.I;          NO WDOT REQUEST--RETURN          2145.000

                                                      IO.WDOT                                                            2146.000

034F 0 5 7 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  306        *LINK CMD.WDOT;                                                    2147.000

0350 9 4 0 0 1 0 0 0 00 0 FB3 0 0 0 0 0 0 0 0  353        IF %IORETRY *GO TO $+3;                                            2148.000

0351 0 5 7 0 1 0 0 0 08 0 00C 0 0 0 0 0 0 0 0  30C        INCRN,*LINK TEST.RETRY;                                            2149.000

0352 0 5 0 0 1 0 0 0 00 0 04F 0 0 0 0 0 0 0 0  34F        *GO TO IO.WDOT;                                                    2150.000

0353 9 4 0 0 1 0 0 0 00 0 1B5 0 0 0 0 0 0 0 0  355        IF %IONORESP:IOTIMEOUT:IOCHBUSY *HOP $+2;                          2151.000

0354 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304        NU=%T(ZE), *GO TO CD.DUD; ERROR OR BUSY EXIT                       2152.000

0355 0 5 0 0 0 0 C 0 00 F 08F 0 0 0 0 0 0 0 0  38F        NU=T(ZE),*GO TO UPDATE.INTERRUPT;                                  2153.000

                                                                                                                         2154.000
                                                      **                                                                 2155.000
                                                      **   ADJUST RIGHT HAND FLAG & EXIT TO INTERRUPT PROCESSING         2156.000
                                                      **                                                                 2157.000

                                                                                                                         2158.000

                                                      IORESPRDY.PENDING                                                  2159.000

                                                      CALM.EXTL                                                          2160.000

0356 0 0 0 0 4 0 0 0 00 0 C00 0 0 0 0 0 0 0 0              NOD=R(OFFSET);                                                     2161.000

                                                           FULLMAR=(EXTPROC1↑20&@FF000000),                              2162.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU NON-CLASS F I/O

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0357 3 6 0 0 1 E E 0 02 A 45D 0 0 0 0 0 0 0 0  45D        IF BMUX18 *GO TO EXTPROC1;  BRANCH IF NOT RHFLAG                  2163.000
0358 0 6 0 0 1 0 0 0 00 C 45D 0 0 0 0 0 0 0 0  45D        SETRHF,*GO TO EXTPROC1;                                          2164.000
                                                  CALM.EXIT                                                             2165.000
0359 0 0 0 0 3 D 8 0 00 0 C00 0 0 0 0 0 0 0 0              R(OFFSET)=S&DT;   CLEAR R(OFFSET)                            2166.000
035A 3 1 0 0 3 D E 0 00 B 004 0 0 0 0 0 0 0 0              FULLMAR=S&DT, RESET(HIREG), IF BMUX19 *JUMPJ; RTN IF NOT CALM 2167.000
035B 5 6 0 0 1 0 0 0 00 A 829 0 0 0 0 0 0 0 0  829        IF %BMUX18 *GO TO FETCH.RETURN;  EXIT IF CALM RHFLAG             2168.000
035C 0 6 0 0 1 0 0 0 00 5 829 0 0 0 0 0 0 0 0  829        TOGRHF, *GO TO FETCH.RETURN; SET RIGHT HAND FLAG & EXIT CALM     2169.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTR.CTL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                2170.000
                                                **                                                              2172.000
                                                **                  INTERRUPT CONTROL INSTRUCTION SEQUENCE      2173.000
                                                **                     ENTRY PARAMETER :                        2174.000
                                                **                       T = CALLER'S REQUEST                   2175.000
                                                **                                                              2176.000
                                                                                                                2177.000
                                                DO.INTR.CTL                                                     2178.000
035D 0 5 7 0 4 1 0 0 00 0 904 0 0 0 0 0 0 0 0  304              NOD=%R(RDEV), *LINK CO.DUO;                     2179.000
035E 6 5 0 0 4 0 0 3 00 0 7A4 0 0 0 0 0 0 0 0  3A4              NOD=ER(INTRTAB), IF ALU4-7Z *GO TO CO.EXIT;      2180.000
035F 5 4 0 0 0 0 8 0 00 4 706 0 0 0 0 0 0 0 0  366              R(BUSREQ)=T, IF %BMUX00 *GOTO ISSUE.INTR.CTL.SEQ; 2181.000
                                                DO.INTR.CTL.BUSY.LOOP                                           2182.000
0360 0 5 7 0 0 0 C 0 00 F 02E 0 0 0 0 0 0 0 0  32E              NU=T(ZE),*LINK RAM.LOAD;                        2183.000
0361 0 0 0 0 4 0 F 0 00 0 700 0 0 0 0 0 0 0 0                   T=R(BUSREQ);  RESTORE CALLER'S REQUEST          2184.000
0362 2 4 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0  366              IF NCTRZ *GO TO ISSUE.INTR.CTL.SEQ;             2185.000
0363 1 4 0 0 1 0 0 0 00 0 BB0 0 0 0 0 0 0 0 0  360              IF IOCHBUSY *GO TO DO.INTR.CTL.BUSY.LOOP;       2186.000
                                                INTR.CTL.ERR                                                    2187.000
0364 0 6 7 0 1 0 0 0 00 0 515 0 0 0 0 0 0 0 0  515              *LINK CLEAR.TIMEOUT;                            2188.000
0365 0 4 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0  367              *GO TO $+2;                                     2189.000
                                                ISSUE.INTR.CTL.SEQ                                              2190.000
0366 0 4 7 0 1 0 0 0 00 0 00A 0 0 0 0 0 0 0 0  36A              *LINK EXECUTE.INTR.CTL;                         2191.000
0367 0 0 0 0 4 0 F 0 00 0 C07 0 0 0 0 0 0 0 0                   T=R(OFFSET), CLRS;                              2192.000
0368 5 5 0 0 0 0 6 0 00 9 059 0 0 0 0 0 0 0 0  359              DT=T, IF %BMUX17 *GO TO CALM.EXIT; TEST FOR INT RTN FLAG 2193.000
0369 0 6 0 0 1 0 0 0 00 0 FB0 0 0 0 0 0 0 0 0  FB0              *GO TO NON.INTERRUPTABLE.EXIT;                  2194.000
                                                                                                                2195.000
                                                **                                                              2196.000
                                                **          EXECUTE INTERRUPT CONTROL                           2197.000
                                                **                                                              2198.000
                                                                                                                2199.000
                                                EXECUTE.INTR.CTL                                                2200.000
036A 0 5 7 0 0 1 C 0 00 F 029 0 0 0 0 0 0 0 0  329              NU=%T(ZE),*LINK COUNT.DOWN;                     2201.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTR.CTL

```
PC   T S M A B + D R  Y X PCH + B D M + R A D  ADDR
036B C 5 0 0 4 0 E 0 00 0 975 0 0 0 0 0 0 0 0  375        FULLMAR=R(RDEV), IF NALUZ *GO TO ISSUE.ICT; BRANCH IF RTOM       2202.000
                                                       RETRY.AICT                                                          2203.000
036C 0 5 7 0 4 0 0 0 00 0 907 0 0 0 0 0 0 0 0  307        NOD=R(RDEV),*LINK CMD.AICT;                                      2204.000
036D 9 4 0 0 1 0 0 0 00 0 5BF 0 0 0 0 0 0 0 0  36F        IF %IONORESP:IOTIMEOUT *HOP $+2;                                 2205.000
036E 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304        NU=%T(ZE), *GO TO CD.DUD;  ERROR EXIT                            2206.000
036F 9 4 0 0 1 0 0 0 00 0 AB2 0 0 0 0 0 0 0 0  372        IF %IORETRY:IOCHBUSY *GO TO INTR.INC;                            2207.000
0370 0 5 7 0 1 0 0 0 08 0 00C 0 0 0 0 0 0 0 0  30C        INCRN,*LINK TEST.RETRY;                                          2208.000
0371 0 5 0 0 1 0 0 0 00 0 06C 0 0 0 0 0 0 0 0  36C        *GO TO RETRY.AICT;                                               2209.000
                                                       INTR.INC                                                            2210.000
0372 9 4 0 0 1 0 0 0 00 0 5A2 0 0 0 0 0 0 0 0  372        IF %IORESPRDY:FFINT *GO TO $;                                    2211.000
0373 1 4 0 0 1 0 0 0 00 0 7A5 0 0 0 0 0 0 0 0  375        IF IURESPRDY *GO TO ISSUE.ICT;                                   2212.000
0374 0 5 0 0 1 0 0 0 00 0 056 0 0 0 0 0 0 0 0  356        *GO TO IURESPRDY.PENDING;                                        2213.000
                                                       ISSUE.ICT                                                           2214.000
0375 0 5 7 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  308        *LINK CMD.ICT;                                                   2215.000
0376 9 4 0 0 1 0 0 0 00 0 5B8 0 0 0 0 0 0 0 0  378        IF %IONORESP:IOTIMEOUT *HOP $+2;                                 2216.000
0377 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304        NU=%T(ZE), *GO TO CD.DUD;  ERROR EXIT                            2217.000
0378 9 4 0 0 1 0 0 0 00 0 ABB 0 0 0 0 0 0 0 0  37B        IF %IORETRY:IOCHBUSY *HOP ICT.DONE;                              2218.000
0379 0 5 7 0 1 0 0 0 08 0 00C 0 0 0 0 0 0 0 0  30C        INCRN,*LINK TEST.RETRY;                                          2219.000
037A 0 4 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0  375        *GO TO ISSUE.ICT;                                                2220.000
                                                       TCT.DONE                                                            2221.000
037B 0 0 0 3 0 D 0 0 02 0 0CD 0 0 0 0 0 0 0 0             NOD=@0000000C&T, CLDNU;        TEST ACTIVATE OR DEACTIVATE       2222.000
037C 0 0 0 3 0 D 0 0 02 0 080 0 0 0 0 0 0 0 0             NOD=@00000008&T;        TEST DEACTIVATE                          2223.000
037D C 4 0 1 1 0 1 0 02 A 85F 0 0 0 0 0 0 0 0  37F        S=SCRATCH(@85), IF NALUZ *HOP $+2; FETCH INTR ACTIVE COUNT        2224.000
037E 0 5 0 0 1 0 0 0 00 0 08F 0 0 0 0 0 0 0 0  38F        *GO TO UPDATE.INTERRUPT; EXIT FOR DI, EI, OR RI                  2225.000
037F C 4 0 0 1 E 0 0 00 0 007 0 0 0 0 0 0 0 0  387        NOD=S,IF NALUZ *GO TO DAI.INSTR;                                 2226.000
                                                       AI.INST                                                             2227.000
0380 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0             NOD=@04000000&FR(OFFSET);    TEST IF LEVEL ACTIVE                2228.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTR.CTL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0381 0 0 0 0 1 6 F 0 00 0 D04 0 0 0 0 0 0 0 0          T=S+1,SET(DINTRA);                                              2229.000
0382 C 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  38F     IF NALUZ *GO TO UPDATE.INTERRUPT;                               2230.000
0383 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  38C     *GO TO UPDATE.ACTIVE;                                           2231.000
                                               CHECK.DRT                                                           2232.000
0384 1 4 0 0 1 0 0 0 00 0 BD4 0 0 0 0 0 0 0 0  384     IF BIBUSY *GO TO S;  WAIT FOR DRT                               2233.000
0385 9 1 0 0 1 0 0 0 00 0 DB0 0 0 0 0 0 0 0 0          IF %IOTIMEOUT *JUMPJ;  RETURN IF NO ERROR (N=0)                 2234.000
0386 0 5 0 0 0 1 C 0 00 F 004 0 0 0 0 0 0 0 0  304     NIL=%T(ZE), *GO TO CD.DUD; SET N=@FF TO INDICATE ERROR          2235.000
                                               DAI.INSTR                                                           2236.000
0387 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0          NOD=@04000000&FR(OFFSET);    TEST IF LEVEL ACTIVE               2237.000
0388 0 0 0 0 1 E 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=S;                                                          2238.000
0389 B 4 0 0 1 4 F 0 00 0 00F 0 0 0 0 0 0 0 0  38F     T=S-1,IF ALUZ *GO TO UPDATE.INTERRUPT;                          2239.000
038A B 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  38F     IF ALUZ *GO TO UPDATE.INTERRUPT;                                2240.000
038B B 0 0 0 1 0 0 0 00 0 504 0 0 0 0 0 0 0 0          IF ALUZ RESET(DINTRA);                                          2241.000
                                               UPDATE.ACTIVE                                                       2242.000
038C 0 4 7 1 1 E 1 0 02 0 854 0 0 0 0 0 0 0 0  384     S=@00850000, *LINK CHECK.DRT;                                   2243.000
038D A 1 0 0 6 0 0 0 00 0 900 0 0 0 0 0 0 0 0          NOD=INTLVL, IF %NCTRZ *JUMPJ;  RETURN IF TIMEOUT ERROR          2244.000
038E 0 5 0 0 0 0 4 0 00 A 090 0 0 0 0 0 0 0 0  390     SCRATCH(S)=T, *GO TO UPDATE.INTERRUPT+1; STORE ACTIVE COUNT      2245.000
                                               UPDATE.INTERRUPT                                                    2246.000
038F 0 5 7 0 1 0 0 0 00 0 084 0 0 0 0 0 0 0 0  384     *LINK CHECK.DRT;                                                2247.000
0390 E 1 0 0 4 0 1 5 00 0 900 0 0 0 0 0 0 0 0          S=FR(INTRLOC), IF %NCTR7 *JUMPJ; TIMEOUT ERROR RETURN           2248.000
0391 0 1 0 0 4 0 4 0 00 A B00 0 0 0 0 0 0 0 0          SCRATCH(S)=R(INTRTAB),*JUMPJ;   STORE INTR TAB ENTRY            2249.000
                                                                                                                   2250.000
                                               **                                                                  2251.000
                                               **      TEST FOR RAM LOADED FLAG IN R(INTRTAB)                      2252.000
                                               **                                                                  2253.000
                                               TEST.RAM.LOAD                                                       2254.000
0392 0 0 0 0 4 0 0 0 00 0 B00 0 0 0 0 0 0 0 0          NOD=R(INTRTAB); TEST FOR RAM LOADED FLAG (BIT 00)               2255.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTR.CTL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0393 5 5 0 0 0 0 8 0 00 4 704 0 0 0 0 0 0 0 0  304        R(BUSREQ)=T, IF %RMUX00 *LEAP CO.DUD; SAVE CALLERS REQUEST      2256.000

0394 0 5 0 0 0 0 C 0 00 F 02E 0 0 0 0 0 0 0 0  32E        NU=T(ZE), *GO TO RAM.LOAD; CLEAR N REG                          2257.000
```

```
                   SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                   CD
PC    T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                   2258.000
                                                        CD                                         2259.000
0395 0 6 7 0 1 0 0 0 00 0 2DB 0 0 0 0 0 0 0 0  2DB          *LINK IO.REG.LOAD;                    2260.000
0396 0 0 0 0 4 1 0 0 00 0 900 0 0 0 0 0 0 0 0               NOD=%R(RDEV); TEST FOR CLASS F DEVICES 2261.000
0397 6 0 0 4 4 D F 1 02 0 7F0 0 0 0 0 0 0 0 0               T=@7FFFFFFF&FR(RDEV); CLEAR PROG VIOLATION FLAG  2262.000
0398 6 5 0 4 0 0 4 0 00 A 7F0 0 0 0 0 0 0 0 0  3F0          SCRATCH(FR(EDEV))=T, IF ALU4-7Z *LEAP EXCLUDE.CD.CLASSF;  2263.000
0399 0 0 0 0 0 D F 0 02 6 0F0 0 0 0 0 0 0 0 0               T=@0F000000&T, SDFST; SAVE CLASS      2264.000
039A 0 0 0 0 0 3 0 0 02 0 FD0 0 0 0 0 0 0 0 0               NOD=@FD000000+T;  TEST FOR CLASS 3 DEVICE  2265.000
039B 0 0 0 0 2 0 8 0 12 F 200 0 0 0 0 0 0 0 0               R(CMD)=IO(SE), SAVESIGN; SAVE CD INSTRUCTION FUNCTION CODE  2266.000
039C B 6 0 0 1 E F 0 00 0 2B5 0 0 0 0 0 0 0 0  2B5          T=S, IF ALUZ *GOTO DEV.CLASS3; FETCH CLASS FROM S TO T  2267.000
039D 0 0 0 2 2 D 0 0 02 0 700 0 0 0 0 0 0 0 0               NOD=@00007000&IO; TEST FOR NON-DATA CD  2268.000
039E 3 5 0 0 1 0 0 0 00 1 0A7 0 0 0 0 0 0 0 0  3A7          IF SIGNSAVE *GOTO CD.CMD;             2269.000
039F B 4 0 0 2 0 0 0 00 0 007 0 0 0 0 0 0 0 0  3A7          NOD=IO,IF ALUZ *GO TO CD.CNTRL;       2270.000
03A0 5 5 0 0 2 0 0 0 00 B 0E5 0 0 0 0 0 0 0 0  3E5          NOD=IO,IF %BMUX19 *GO TO CD.TERMINATE;  2271.000
03A1 5 5 0 0 4 0 F 0 00 A 9F3 0 0 0 0 0 0 0 0  3F3          T=R(RDEV), IF %BMUX18 *GO TO CD.TCWA;  2272.000
                                                        PROG.VIOLATION                             2273.000
03A2 6 0 0 0 4 A F 1 02 0 800 0 0 0 0 0 0 0 0               T=@80000000:FR(RDEV);  SET PROG VIOL  2274.000
03A3 0 0 0 4 0 0 4 0 00 A 800 0 0 0 0 0 0 0 0               SCRATCH(R(EDEV))=T;                   2275.000
                                                        CD.EXIT                                    2276.000
03A4 9 4 0 0 1 0 0 0 00 0 5B6 0 0 0 0 0 0 0 0  3A6          IF %IONORESP:IOTIMEOUT *GO TO $+2;    2277.000
03A5 0 6 7 0 1 0 0 0 00 0 515 0 0 0 0 0 0 0 0  515          *LINK CLEAR.TIMEOUT;                  2278.000
03A6 0 3 0 0 1 0 0 0 19 0 084 0 0 0 0 0 0 0 0               FETCHPC,*JUMPZ,RESET(HIREG);          2279.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    CD
PC  TSMAB+DR YX PCH+BDM+BAD ADDR

                                                                                                                    2280.000
                                        **                                                                          2282.000
                                        **                    INPUT/OUTPUT CD                                       2283.000
                                        **                                                                          2284.000
                                        **                                                                          2285.000

                                                                                                                    2286.000
                                        CD.CMD                                                                      2287.000

                                        CD.CNTRL                                                                    2288.000

03A7 0 0 0 0 4 0 E 0 00 0 000 0 0 0 0 0 0 0 0           FULLMAR=R(IOCD.ADDR);  FETCH IOCD ADDR                      2289.000

03A8 6 0 0 0 4 D F 1 02 0 0F0 0 0 0 0 0 0 0 0           T=@0F000000&FR(RDEV); GET DVC CLASS                         2290.000

03A9 0 0 0 0 0 8 0 0 02 0 0E0 0 0 0 0 0 0 0 0           NOD=@0E000000!T; TEST FOR CLASS =F                          2291.000

03AA 0 0 0 0 0 3 0 0 02 0 FD0 0 0 0 0 0 0 0 0           NOD=@FD000000+T; TEST FOR CLASS >2                          2292.000

03AB B 5 0 0 2 0 0 0 00 0 0D7 0 0 0 0 0 0 0 0 3D7       NOD=IO, IF ALUZ *GO TO DEV.CLASSE;                          2293.000

03AC 5 4 0 0 2 0 0 0 00 2 002 0 0 0 0 0 0 0 0 3A2       NOD=IO, IF %ALUNEGW *GO TO PROG.VIOLATION; EXIT IF CLASS >2 2294.000

                                        DEV.CLASS.012                                                               2295.000

03AD 3 4 0 0 1 E 1 0 02 B 01F 0 0 0 0 0 0 0 0 3AF       S=@01000000, IF HMUX19 *GO TO $+2; SET IOCD CMD TO WRITE    2296.000

03AE 0 0 0 0 1 E 1 0 02 0 020 0 0 0 0 0 0 0 0           S=@02000000; SET IOCD CMD TO READ                           2297.000

03AF 0 0 0 0 0 3 C 0 02 0 FF0 0 0 0 0 0 0 0 0           NU=@FF000000+T; TEST FOR CLASS 1                            2298.000

03B0 0 0 0 0 1 E 8 0 00 0 200 0 0 0 0 0 0 0 0           R(CMD)=S; SAVE IOCD CMD                                     2299.000

03B1 B 4 0 0 2 0 0 0 00 0 00D 0 0 0 0 0 0 0 0 3BD       NOD=IO, IF ALUZ *GO TO DEV.CLASS1;                          2300.000

03B2 A 5 0 0 2 0 0 0 00 0 0C4 0 0 0 0 0 0 0 0 3C4       NOD=IO, IF %NCTR0 *GO TO DEV.CLASS2; (BRANCH NOT NEGATIVE)  2301.000

                                                                                                                    2302.000
                                        **                                                                          2303.000
                                        **                    DEVICE CLASS 0--LINE PRINTER                          2304.000
                                        **                                                                          2305.000

                                                                                                                    2306.000
                                        DEV.CLASS0                                                                  2307.000

03B3 0 0 0 0 2 0 8 0 00 0 400 0 0 0 0 0 0 0 0           R(IOCD1)=IO;              LOAD INSTR                        2308.000

03B4 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0           T=DI;                                                       2309.000

03B5 0 0 0 0 5 0 1 0 00 0 400 0 0 0 0 0 0 0 0           S=R(IOCD1,HWS);                                             2310.000

03B6 0 0 0 0 2 1 E 6 0 00 0 000 0 0 0 0 0 0 0 0         DT=SNIBL;                                                   2311.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
SEL  32/75  CPU                                                    CD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

03B7 0 0 0 0 1 0 0 0 01 3 020 0 0 0 0 0 0 0 0          SHIFTD(SLLD);                                          2312.000

03B8 0 0 0 0 1 0 0 0 00 3 080 0 0 0 0 0 0 0 0          SHIFTDI(SRL);                                          2313.000

03B9 0 0 0 0 1 0 0 0 01 3 080 0 0 0 0 0 0 0 0          SHIFTD(SRLD);                                          2314.000

03BA 0 0 0 0 3 0 1 0 02 0 880 0 0 0 0 0 0 0 0          S=@R8000000&DI;                                        2315.000

03BB 0 0 0 0 0 0 6 0 00 0 000 0 0 0 0 0 0 0 0          DT=T;                                                  2316.000

03BC 0 5 0 0 1 E 8 0 00 0 4CA 0 0 0 0 0 0 0 0  3CA     R(IOCD1)=S, *GO TO BUILD.IOCD;                         2317.000

                                                                                                          2318.000
                                                ;;                                                        2319.000
                                                ;;          DEVICE CLASS 1--CARD READER                   2320.000
                                                ;;                                                        2321.000

                                                                                                          2322.000

                                                DEV.CLASS1                                                2323.000

03BD 0 0 0 2 2 0 F 0 02 0 0C0 0 0 0 0 0 0 0 0          T=@00000C00&IO;      COPY BITS 20-21 OF FUNCTION CODE  2324.000

03BE 0 0 0 0 1 E 1 0 02 0 200 0 0 0 0 0 0 0 0          S=@20000000;           SET HALF ASCII                 2325.000

03BF B 4 0 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0  3C3     IF ALUZ *GO TO DEV.CLASS1.EXIT;                        2326.000

03C0 0 0 0 2 0 0 0 0 02 0 040 0 0 0 0 0 0 0 0          NOD=@00000400&T;      TEST BIT 21                      2327.000

03C1 0 0 0 0 1 E 1 0 02 0 080 0 0 0 0 0 0 0 0          S=@08000000;           SET AUTO MODE                  2328.000

03C2 B 0 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0          IF ALUZ CLRS;          SET BINARY MODE                2329.000

                                                DEV.CLASS1.EXIT                                           2330.000

03C3 0 4 0 0 1 E 8 0 00 0 40A 0 0 0 0 0 0 0 0  3CA     R(IOCD1)=S, *GO TO BUILD.IOCD;                         2331.000

                                                                                                          2332.000
                                                ;;                                                        2333.000
                                                ;;          DEVICE CLASS 2--TELETYPE                      2334.000
                                                ;;                                                        2335.000

                                                                                                          2336.000

                                                DEV.CLASS2                                                2337.000

03C4 3 4 0 0 1 0 0 0 00 9 007 0 0 0 0 0 0 0 0  3C7     IF BMUX17 *GO TO $+3;  TEST FUNC CODE BIT 17           2338.000

                                                FORCE.PROG.VIOL                                           2339.000

03C5 0 0 0 0 1 E 1 0 02 0 F00 0 0 0 0 0 0 0 0          S=@F0000000;                                           2340.000

03C6 0 4 0 0 1 E 8 0 00 0 203 0 0 0 0 0 0 0 0  3C3     R(CMD)=S, *GO TO DEV.CLASS1.EXIT;   CLR RD/WR BIT      2341.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                                  C D
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

03C7 0 0 0 0 1 E 1 0 02 0 080 0 0 0 0 0 0 0 0          S=@08000000;            SET ECHO                                  2342.000

03C8 0 0 0 2 2 D 0 0 02 0 080 0 0 0 0 0 0 0 0          NOD=@00000800&IO;       TEST FUNCTION CODE BIT 20                 2343.000

03C9 0 4 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0  3C2     *GO TO DEV.CLASS1.EXIT-1;                                 2344.000

                                                                                                                  2345.000
                                                ;;                                                                2346.000
                                                ;;      BUILD IOCD FOR DEVICE CLASS 0, 1, AND 2 (TLC DEVICES)     2347.000
                                                ;;                                                                2348.000

                                                                                                                  2349.000

                                                BUILD.IOCD                                                        2350.000

03CA 0 5 7 0 1 E E 0 02 0 312 0 0 0 0 0 0 0 0  312     FULLMAR=(FORMAT.TCW↑20&@FF000000), *LINK FORMAT.TCW; CLEAR MAR  2351.000

                                                                                                                  2352.000
                                                ;;                                                                2353.000
                                                ;;        RETURN PARAMETERS :                                     2354.000
                                                ;;                R(IOCD1) = TRANSLATED IOCD CMD (IOCD WORD 1)    2355.000
                                                ;;                R(D.ADDR) = I/O DATA ADDRESS (IOCD WORD 2)      2356.000
                                                ;;                R(IOCD.ADDR) = IOCD ADDRESS                     2357.000
                                                ;;                R(TCW) = IOCD INPUT/OUTPUT DATA BYTE COUNT      2358.000
                                                ;;                R(CMD) = READ/WRITE IOCD CMD                    2359.000
                                                ;;                                                                2360.000

                                                                                                                  2361.000

03CB 0 0 0 0 2 0 0 0 00 E 000 0 0 0 0 0 0 0 0          NOD=IO(SE);                  TEST INPUT/OUTPUT                    2362.000

03CC 0 0 0 3 1 E F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004;            CONSTANT 4                                2363.000

03CD 5 5 0 4 0 3 E 0 00 2 0D2 0 0 0 0 0 0 0 0  3D2     FULLMAR=R(IOCD.ADDR)+T,TF %ALUNFGW *GO TO SINGLE.CTL.IOCD;  2364.000

                                                ;;                                                                2365.000

                                                ;;              BUILD I/O IOCD                                    2366.000

                                                ;;                                                                2367.000

                                                SINGLE.IOCD                                                       2368.000

03CE 0 0 0 0 4 0 F 0 1E 1 180 0 0 0 0 0 0 0 0          T=R(D.ADDR), WRTIF, FRCWORD;   WRTIF IOCD(32-63)           2369.000

03CF 0 0 0 0 4 0 1 0 00 0 400 0 0 0 0 0 0 0 0          S=R(IOCD1);         READ TRANSLATED IOCD CMD              2370.000

03D0 0 0 0 0 4 A 1 0 00 0 200 0 0 0 0 0 0 0 0          S=S:R(CMD);               COMBINE TRANSLATED/RD-WT IOCD CMD   2371.000

03D1 0 4 0 0 4 A F 0 00 0 305 0 0 0 0 0 0 0 0  3D5     T=S:R(TCW),*GO TO STORE.IOCD0;                            2372.000

                                                ;;                                                                2373.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                                    C D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                    **              BUILD CONTROL IOCD                                  2374.000
                                                    **                                                                  2375.000
                                                    SINGLE.CTL.IOCD                                                     2376.000
03D2 0 5 7 0 4 0 F 0 00 0 F00 0 0 0 0 0 0 0 0  300          T=R(ZERO),*LINK MEMORY.WRITE;          CLEAR IOCD (32-63)  2377.000
03D3 0 0 0 0 1 6 F 0 02 0 060 0 0 0 0 0 0 0 0               T=@06000000+1;       CONTROL IOCD CMD AND BYTE COUNT OF 1  2378.000
03D4 0 0 0 4 0 A F 0 00 0 400 0 0 0 0 0 0 0 0               T=R(IOCD1):T;                                              2379.000
                                                    STORE.IOCD0                                                         2380.000
03D5 0 5 7 0 4 0 E 0 00 0 000 0 0 0 0 0 0 0 0  300          FULLMAR=R(IOCD.ADDR),*LINK MEMORY.WRITE;                   2381.000
03D6 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  30C          *GO TO START.IOC;                                          2382.000
                                                                                                                       2383.000
                                                    **                                                                  2384.000
                                                    **       BUILD IOCD FOR DEVICE CLASS F                              2385.000
                                                    **                                                                  2386.000
                                                    **       ENTRY PARAMETERS :                                         2387.000
                                                    *          R(IOCD.ADDR) = IOCD ADDR                                 2388.000
                                                    *          FULLMAR = IOCD ADDRESS                                   2389.000
                                                    *          R(CMD) = CD BITS 16 - 31                                 2390.000
                                                    *          R(TI) = TCW ADDR                                         2391.000
                                                                                                                       2392.000
                                                    DEV.CLASSF                                                          2393.000
03D7 0 0 0 0 4 0 F 0 1F F 280 0 0 0 0 0 0 0 0               T=R(CMD,ZE), WRITE; FETCH IOCD WORD 0 & STORE IN MEMORY    2394.000
03D8 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0               *NOP; WAIT FOR MEMORY                                      2395.000
03D9 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0               FULLMAR=@00000004+MAR; SET IOCD WORD 1 ADDRESS             2396.000
03DA 0 0 0 0 4 0 F 0 1F 1 680 0 0 0 0 0 0 0 0               T=R(TI), WRITE, FRCWORD; STORE IOCD WORD 1 IN MEMORY        2397.000
03DB 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0               *NOP;  WAIT FOR MEMORY                                     2398.000
                                                    START.IOC                                                           2399.000
03DC 6 0 0 0 4 A B 3 02 0 400 0 0 0 0 0 0 0 0               FR(INTRTAB)=@40000000:FR(INTRTAB);      FLAG IO STARTED    2400.000
03DD 0 0 0 0 1 E F 0 02 0 800 0 0 0 0 0 0 0 0               T=@80000000;              START I/O REQUEST                2401.000
03DE 0 5 7 4 0 A F 0 00 0 023 0 0 0 0 0 0 0 0  323          T=R(IOCD.ADDR):T, *LINK CMD.IOC;   SET SIO BIT IN WDOT DATA  2402.000
                                                                                                                       2403.000
                                                    ****     RETURN PARAMETERS                                          2404.000
                                                    **         NCTR = 00 IF NO ERRORS                                   2405.000
                                                    **         DI BIT 16 =DVC BUSY                                      2406.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    CD
PC  TSMAB+DR YX PCH+BDM+BAD ADDR
                                                                                                    2407.000
03DF 2 4 0 0 3 0 0 0 00 0 902 0 0 0 0 0 0 0 0 3E2       NOD=DI, IF NCTR7 *GO TO CD.NORMAL.EXIT-1; TEST FOR I/O ERRORS      2408.000
03E0 1 4 0 0 1 0 0 0 00 0 5B4 0 0 0 0 0 0 0 0 3E4       IF IONORESP:IOTIMEOUT *HOP CD.ERROR.EXIT;                          2409.000
03E1 0 5 0 0 1 0 0 0 00 0 0A2 0 0 0 0 0 0 0 0 3A2       *GO TO PROG.VIOLATION;  DVC BUSY PROGRAM VIOLATION                 2410.000
03E2 5 5 0 0 3 0 0 0 00 8 0A2 0 0 0 0 0 0 0 0 3A2       NOD=DI, IF %BMUX16 *GO TO PROG.VIOLATION; DVC BUSY EXIT            2411.000
                                                CD.NORMAL.EXIT                                      2412.000
03E3 0 3 0 0 1 0 0 0 19 0 084 0 0 0 0 0 0 0 0           FETCHPC, RESET(HIREG), *JUMPZ; NO ERRORS EXIT PATH                 2413.000
                                                CD.ERROR.EXIT                                       2414.000
03E4 0 5 0 0 1 0 0 0 00 0 0A4 0 0 0 0 0 0 0 0 3A4       *GO TO CD.EXIT;                                                    2415.000
                                                                                                    2416.000
                                                **                                                  2417.000
                                                **                  TERMINATE CD REQUEST            2418.000
                                                **                                                  2419.000
                                                                                                    2420.000
                                                CD.TERMINATE                                        2421.000
03E5 0 5 7 0 0 1 C 0 00 F 092 0 0 0 0 0 0 0 0 392       NU=%T(ZE),*LINK TEST.RAM.LOAD;                                     2422.000
03E6 0 0 0 0 4 0 E 0 00 0 900 0 0 0 0 0 0 0 0           FULLMAR=R(RDEV);                                                   2423.000
03E7 0 0 0 0 1 E F 0 02 0 200 0 0 0 0 0 0 0 0           T=@20000000;            HALT REQUEST                               2424.000
03E8 0 0 0 0 1 0 0 0 1E 0 000 0 0 0 0 0 0 0 0           WDOT;                                                              2425.000
03E9 0 0 0 0 4 0 E 0 00 0 902 0 0 0 0 0 0 0 0           FULLMAR=R(RDEV),FORCE7;                                            2426.000
03EA 0 5 7 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0 303       *LINK TEST.BIBUSY;                                                 2427.000
03EB 1 4 0 0 1 0 0 0 00 0 5BF 0 0 0 0 0 0 0 0 3EF       IF IONORESP:IOTIMEOUT *GO TO CD.2ND.TERMINATE+1;                   2428.000
03EC 9 4 0 0 1 0 0 0 00 0 5AC 0 0 0 0 0 0 0 0 3EC       IF %IORESPRDY:FFINT *GO TO $;                                      2429.000
03ED 3 5 0 0 1 0 0 0 00 0 056 0 0 0 0 0 0 0 0 356       IF EXTLW *GOTO IORESPRDY.PENDING ;                                 2430.000
                                                CD.2ND.TERMINATE                                    2431.000
03EE 0 5 7 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0 306       *LINK CMD.WDOT;                                                    2432.000
03EF 0 5 0 0 1 0 0 0 00 0 0A4 0 0 0 0 0 0 0 0 3A4       *GO TO CD.EXIT;                                                    2433.000
                                                EXCLUDE.CD.CLASSF                                   2434.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                                  C D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

03F0 0 0 0 1 1 E F 0 02 6 080 0 0 0 0 0 0 0 0                T=UNDEF.75.FLG, SDEST; SET SYSTEM CHECK TRAP ERROR CODE          2435.000

03F1 A 5 0 0 1 0 0 0 00 0 3A4 0 0 0 0 0 0 0 0  3A4           IF %MODE75 *GO TO CD.EXIT;                                       2436.000

03F2 0 6 0 0 1 0 0 0 00 0 57C 0 0 0 0 0 0 0 0  57C           *GO TO SYSTEM.CHECK.TRAP;                                        2437.000

                                                                                                                              2438.000
                                                     **                                                                       2439.000
                                                     **                                                                       2440.000
                                                     **                                                                       2441.000
                                                     **                  TRANSFER CURRENT WORD ADDRESS                        2442.000
                                                     **                                                                       2443.000

                                                                                                                              2444.000

                                                     CD.TCWA                                                                  2445.000

03F3 0 0 0 4 0 0 4 0 00 A 800 0 0 0 0 0 0 0 0                SCRATCH(K(EDEV))=T;     RESTORE ORIGINAL DEV STAT                2446.000

03F4 0 4 7 3 1 E F 0 02 0 088 0 0 0 0 0 0 0 0  3F8           T=@00000008, *LINK CD.TCWA.IO;                                   2447.000

03F5 2 4 0 0 3 0 0 0 00 0 909 0 0 0 0 0 0 0 0  3F9           NOD=DI,IF NCTRZ *GO TO $+4;                                      2448.000

03F6 1 4 0 0 1 0 0 0 00 0 BB4 0 0 0 0 0 0 0 0  3F4           IF TOCHBUSY *GO TO $-2;                                          2449.000

                                                     TCWA.ERROR.EXIT                                                          2450.000

03F7 0 5 0 0 1 0 0 0 00 0 0A4 0 0 0 0 0 0 0 0  3A4           *GO TO CD.EXIT;                                                  2451.000

                                                     CD.TCWA.IO                                                               2452.000

03F8 0 5 0 0 1 0 0 0 00 0 023 0 0 0 0 0 0 0 0  323           *GO TO CMD.IOC;                                                  2453.000

03F9 5 4 0 0 1 0 0 0 00 8 004 0 0 0 0 0 0 0 0  3F4           IF %BMUX16 *GO TO CD.TCWA+1;                                     2454.000

03FA 0 0 0 3 1 E F 0 02 0 020 0 0 0 0 0 0 0 0                T=@00000002;            TRANSFER COUNT                           2455.000

03FB 0 4 7 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  3F8           *LINK CD.TCWA.IO;       CHECK BUSY                               2456.000

03FC 2 4 0 0 1 0 0 0 00 0 90F 0 0 0 0 0 0 0 0  3FF           IF NCTRZ *GO TO $+3;                                             2457.000

03FD 1 4 0 0 1 0 0 0 00 0 BBB 0 0 0 0 0 0 0 0  3FB           IF TOCHBUSY *HOP $-2;                                            2458.000

03FE 0 5 0 0 1 0 0 0 00 0 0A4 0 0 0 0 0 0 0 0  3A4           *GO TO CD.EXIT;                                                  2459.000

03FF 6 0 0 0 4 8 0 1 02 0 0E0 0 0 0 0 0 0 0 0                NOD=@0E000000!FR(RDEV); TEST CLASS F                             2460.000

0400 0 0 0 0 3 0 8 0 00 F 000 0 0 0 0 0 0 0 0                R(TMP0)=DI(ZE); SAVE TCWA STATUS                                 2461.000

0401 2 4 0 0 3 0 1 0 00 0 70D 0 0 0 0 0 0 0 0  40D           S=DI, IF ALU4-7Z *GO TO STORE.CNT; SAVE TCA STATUS & BRANCH ON E 2462.000

                                                     CLASS.012                                                                2463.000

0402 0 6 7 0 1 E E 0 02 0 312 0 0 0 0 0 0 0 0  312           FULLMAR=(FORMAT.TCW!20&@FF000000), *LINK FORMAT.TCW; CLEAR MAR   2464.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    CD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0403 0 0 0 0 4 0 1 0 00 0 300 0 0 0 0 0 0 0 0                    S=R(TCW);                                                    2465.000
0404 0 0 0 0 4 5 F 0 00 0 000 0 0 0 0 0 0 0 0                    T=S-R(TMP0);          BTRANSFERED COUNT IN T                 2466.000
0405 0 0 0 4 0 3 6 0 00 0 100 0 0 0 0 0 0 0 0                    DI=R(D.ADDR)+T;           UPDATE ADDR                        2467.000
0406 0 0 0 4 0 5 F 0 00 0 300 0 0 0 0 0 0 0 0                    T=R(TCW)-T;            DECREMENT TCA COUNT                   2468.000
0407 0 0 0 0 1 0 0 0 00 2 000 0 0 0 0 0 0 0 0                    TNIPL;                                                       2469.000
0408 0 5 7 0 0 0 8 0 00 F 486 0 0 0 0 0 0 0 0  486               R(TMP4)=T(ZE),*LINK DUD.INTR;                                2470.000
0409 0 0 0 0 5 0 1 0 00 0 400 0 0 0 0 0 0 0 0                    S=R(TMP4,HWS);          SHIFT COUNT                          2471.000
040A 5 4 0 0 3 A 1 0 00 F 00D 0 0 0 0 0 0 0 0  400               S=S:DI,IF %BYTE *GO TO STORE.CNT;          UPDATED TCA IN S  2472.000
040B 0 0 0 1 1 E F 0 02 0 080 0 0 0 0 0 0 0 0                    T=@00080000;   F-BIT                                         2473.000
040C 0 0 0 0 0 A 1 0 00 0 000 0 0 0 0 0 0 0 0                    S=S:T;             FOLD IN F-BIT                             2474.000
                                                       STORE.CNT                                                          2475.000
040D 0 0 0 0 4 0 E 0 00 0 600 0 0 0 0 0 0 0 0                    FULLMAR=R(TI);  FETCH TCW ADDR                               2476.000
040E 0 0 0 0 1 E F 0 1F 1 080 0 0 0 0 0 0 0 0                    T=S, WRITE, FRCWORD;  STORE CURRENT TCWA                     2477.000
040F 0 6 0 0 1 0 0 0 00 0 3E3 0 0 0 0 0 0 0 0  3E3               *GO TO CD.NORMAL.EXIT;                                       2478.000
                                                                                                                          2479.000
                                                       ***                                                                2480.000
                                                       ***   LOAD MAP INSTRUCTION, -LMAP-, SECONDARY DECODE ***           2481.000
                                                       ***   (Q=0B), OP CODE = 2C ***                                     2482.000
                                                       ***                                                                2483.000
                                                       ***   ENTRY PARAMETERS:                                            2484.000
                                                       ***     CPU MUST BE UNMAPPED.                                      2485.000
                                                       ***     R(R) = ADDRESS OF A PROGRAM STATUS DOUBLE WORD              2486.000
                                                       ***                                                                2487.000
                                                                                                                          2488.000
                                                       LMAP                                                               2489.000
0410 0 6 7 0 1 0 0 0 0C 0 7F0 0 0 0 0 0 0 0 0  7F0               RSTRHF, *LINK CHECK.CPU.MODE; GO CHECK 75 MODE                2490.000
0411 9 4 0 0 1 0 0 0 00 0 F73 0 0 0 0 0 0 0 0  413               IF %MAPMODE *HOP $+2; LMAP IS ILLEGAL IN MAP MODE            2491.000
0412 0 6 0 0 1 0 0 0 00 0 F33 0 0 0 0 0 0 0 0  F33               *GO TO MAP.MODE.INVALID.ERR+1; EXIT TO SYSTEM CHECK TRAP     2492.000
0413 0 0 0 3 4 3 E 4 02 0 040 0 0 0 0 0 0 0 0                    FULLMAR=@00000004+R(R); GENERATE PSW2 ADDRESS                2493.000
0414 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B               *LINK MEMORY.READ;  GO FETCH PSW 2                           2494.000
0415 0 0 0 0 3 0 8 0 00 0 000 0 0 0 0 0 0 0 0                    R(N.PSW2)=DI;   SAVE NEW PSD 2                               2495.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                                              C D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0416 0 6 7 0 0 0 0 0 00 D 9D3 0 0 0 0 0 0 0 0  9D3      NOD=T, *LINK EVALUATE.PSD2.MAP, OTHERBANK;GO IMPLEMENT PSD2      2496.000
0417 0 0 0 0 4 0 F 0 00 D 000 0 0 0 0 0 0 0 0           T=R(N.PSW2), OTHERBANK;                                          2497.000
0418 0 0 0 4 0 D F 0 02 0 3F5 0 0 0 0 0 0 0 0           T=@3FFFFFFF&T, RESET(MAPMODE); RESET GRANULARITY BITS            2498.000
0419 0 0 0 0 0 0 E 0 00 F 000 0 0 0 0 0 0 0 0           FULLMAR=T(ZE); CLEAR UPPER MAR (RESET 'MAPSET' F/F)              2499.000
041A 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0           FETCHPC;                                                         2500.000
041B 0 3 0 1 0 0 4 0 02 A 900 0 0 0 0 0 0 0 0           SCRATCH(@90)=T, *JUMP7;                                          2501.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    BRI
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2502.000

                                                                                                                    2503.000
                                                    **                                                              2504.000
                                                    **                                                              2505.000
                                                    **      BRANCH AND RESET INTERRUPT (BRI), SECONDARY DECODE      2506.000
                                                    **       (Q=3E), OP CODE = F8                                   2507.000
                                                    **                                                              2508.000

                                                                                                                    2509.000

                                                    (@41C)                                                          2510.000

                                                    BRI                                                             2511.000

041C 0 6 0 0 1 0 0 0 00 0 CEA 0 0 0 0 0 0 0 0  CEA          *GOTO IPU.BRI;            CHECK FOR IPU MODE          2512.000

                                                    *                                                               2513.000

                                                    BRI.RTN                                                         2514.000

                                                    *                                                               2515.000

041D 9 4 0 0 0 0 C 0 00 F DAF 0 0 0 0 0 0 0 0  41F          NU=T(ZE),IF %FFINT *GO TO $+2; SEE IF POLLING LVL IS REQUESTING 2516.000

                                                    BRI.EXIT                                                        2517.000

041E 0 5 0 0 1 0 0 0 00 0 050 0 0 0 0 0 0 0 0  450          *GO TO EXTPROC1; HANDLE INTERRUPT BEFORE BRT               2518.000

041F 1 4 0 0 1 0 0 0 00 0 0F1 0 0 0 0 0 0 0 0  421          IF INTRENA *GO TO $+2;  SEE IF INTERRUPTS ARE ENABLED      2519.000

0420 6 0 0 1 4 0 1 4 02 0 7F0 0 0 0 0 0 0 0 0                S=@007F0000&FR(ALEVEL); IF INTERRUPTS NOT ENABLED USE      2520.000

                                                    *                    THE COMPUTED HIGHEST ACTIVE LEVEL.         2521.000

0421 9 4 0 4 2 3 7 2 00 0 B83 0 0 0 0 0 0 0 0  423          MARIX=R(X)+I0, IF %PRIVBIT *GO TO $+2;                     2522.000

0422 0 6 0 0 1 0 0 0 00 0 9AC 0 0 0 0 0 0 0 0  9AC          *GO TO PRIV.VIOL; EXIT TO PRIVLEGED ERROR                  2523.000

0423 0 0 0 0 1 0 F 0 1D 1 884 0 0 0 0 0 0 0 0                T=MAR, FETCHV, FRCWORD, SET(HIREG); FETCH INSTRUCTION OR   2524.000

                                                    *          INDIRECT WORD (PSW 1) AND SAVE NCTR & MAR FOR NOT-INDR FLAGS 2525.000

0424 2 4 0 0 1 E C 0 02 7 C76 0 0 0 0 0 0 0 0  426          NU=@C7000000,I1TOI0,IF INDIR *GO TO BRI0; FLAG POSSIBLE INDIRECT 2526.000

0425 0 4 0 0 0 0 8 0 00 0 508 0 0 0 0 0 0 0 0  42B          R(BRI.INDR)=T, *GOTO BRI1; SAVE NCTR (NOT INDIRECT FLAG) AND MAR 2527.000

                                                    BRI0                                                            2528.000

0426 0 0 0 0 3 0 0 0 00 0 000 0 0 0 0 0 0 0 0                NOD=DI;  WAIT FOR DRT                                      2529.000

0427 0 5 7 0 1 0 8 0 00 0 486 0 0 0 0 0 0 0 0  486          R(PSD.ADDR)=N, *LINK DUD.INTR; SAVE NCTR(BRI-INDIR FLAG) & MAR 2530.000

0428 2 5 0 4 3 3 7 1 00 0 11E 0 0 0 0 0 0 0 0  41E          MARIX=R(DIX)+DI,OTHERBANK,IF EXTL *GO TO BRI.EXIT;         2531.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                          B R I
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0429 0 0 0 0 1 0 8 0 1D 1 580 0 0 0 0 0 0 0 0          R(BRI.INDR)=MAR, FETCHV, FRCWORD; SAVE NCTR(BRI-INDIR FLAG)&MAR  2532.000
042A 2 4 0 0 1 E C 0 02 0 C76 0 0 0 0 0 0 0 0  426     NU=@C7000000,IF INDIR *GO TO BRI0;                              2533.000
                                                BRI1                                                                    2534.000
042B 0 0 0 0 3 0 8 0 00 0 200 0 0 0 0 0 0 0 0          R(PSW1)=DI;                                                      2535.000
042C 0 5 7 0 1 E 8 0 00 0 A67 0 0 0 0 0 0 0 0  467     R(INTR)=S ,*LINK BUILD.INTR.ADDR;                                2536.000
042D 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0          NOD=@04000000&FR(OFFSET);    TEST IF ACTIVE LEVEL                2537.000
042E 6 0 0 4 4 D B 3 02 0 E30 0 0 0 0 0 0 0 0          FR(INTRTAB)=@E3FFFFFF&FR(INTRTAB); RST ALEXP,IGNSTOP,ACTIVE     2538.000
042F B 4 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0  437     IF ALUZ *GO TO BRI.JUMP; BRANCH IF LEVEL IS NOT ACTIVE           2539.000
0430 0 0 0 0 4 1 0 0 00 0 900 0 0 0 0 0 0 0 0          NOD=%R(RDEV); TEST FOR CLASS F INTERRUPT LEVEL                   2540.000
                                                *             IS NOT ACTIVE AND TEST FOR CLASS F CHANNEL                 2541.000
                                                BRI.DAI                                                                 2542.000
0431 0 0 0 3 1 E F 0 02 0 080 0 0 0 0 0 0 0 0          T=@00000008;            DEACTIVATE INTERRUPT REQUEST                    2543.000
0432 A 4 0 0 1 0 0 0 00 0 704 0 0 0 0 0 0 0 0  434     IF %ALU4-7Z *GO TO $+2; BRANCH IF NOT CLASS F                   2544.000
0433 0 6 0 0 1 0 0 0 00 0 E16 0 0 0 0 0 0 0 0  E16     *GO TO DISPATCH.EXT.DEACTIVATE; GO HANDLE CLASS F CHANNEL       2545.000
0434 0 6 7 0 1 0 0 0 00 0 36A 0 0 0 0 0 0 0 0  36A     *LINK EXECUTE.INTR.CTL;                                         2546.000
0435 2 4 0 0 1 0 0 0 00 0 907 0 0 0 0 0 0 0 0  437     IF NCTRZ *GO TO BRI.JUMP;                                       2547.000
0436 0 6 7 0 0 1 C 0 00 F 58E 0 0 0 0 0 0 0 0  58E     NU=%T(ZE), *LINK BRI.ERROR; GO HANDLE BRI I/O ERROR             2548.000
                                                BRI.JUMP                                                                2549.000
0437 0 0 0 0 4 0 E 0 00 0 50D 0 0 0 0 0 0 0 0          FULLMAR=R(BRI.INDR),CLONU; LOAD NCTR & MAR FOR 55 MODE BRI       2550.000
0438 6 0 0 0 4 0 1 1 00 D 004 0 0 0 0 0 0 0 0          S=FR(STMASK),OTHERBANK, RESET(HIREG); S=@FE000000 OR @FC000000   2551.000
0439 6 4 0 4 1 D 2 0 00 D 90F 0 0 0 0 0 0 0 0  43F     PC=FR(PCMASK)&MAR, OTHERBANK, IF NCTR7 *GO TO BRI.LDPC;          2552.000
                                                *        LOAD PC FOR 55 MODE BRI AND BRANCH IF BRI INDIRECT              2554.000
                                                                                                                        2555.000
                                                ***                                                                     2556.000
                                                ***    BRI INDIRECT ALTER PSW 1 (PRIV BIT, CC'S AND CLEAR ERRORS)       2557.000
                                                ***                                                                     2558.000
                                                                                                                        2559.000
043A 0 0 0 0 4 0 F 0 00 D 204 0 0 0 0 0 0 0 0          T=R(PSW1),OTHERBANK, RESET(PRIV); GET PSW 1 & ENTER PRIV MODE    2560.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU BRI

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0438 0 0 0 0 0 D 1 0 00 0 404 0 0 0 0 0 0 0 0          S=S&T,RESET(DPEFF); LOAD NEW CC'S & EXT BIT IN S                2561.000
043C 0 0 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0          SETXCC(S);   RESTORE CC'S , AND EXTENDED INDEXING               2562.000
043D 5 0 0 0 1 0 0 0 00 2 A04 0 0 0 0 0 0 0 0          IF %ALUNEGW SET(PRIV); ENTER UNPRIVILGED MODE IF NO PRIV BIT    2563.000
043E 4 6 0 0 1 0 0 0 00 C 2C4 0 0 0 0 0 0 0 0  2C4     IF MODE75S *GO TO BRI.FORMAT.PSW; IF 75 MODE GO HANDLE PSD      2564.000
                                                BRI.LDPC                                                               2565.000
043F 0 0 0 0 6 0 8 0 00 0 C00 0 0 0 0 0 0 0 0          R(ALEVEL)=INTLVL;   SAVE NEW ACTIVE LEVEL                       2566.000
0440 0 6 0 0 1 0 0 0 00 0 936 0 0 0 0 0 0 0 0  936     *GO TO BR3; 55 MODE BRI EXIT                                    2567.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU EI,DI,RI,DAT,AI

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                2568.000
                                                    **                                                          2569.000
                                                    **          ENABLE INTERRUPT INSTRUCTION                    2570.000
                                                    **                                                          2571.000
                                                    EI                                                          2572.000
0441 0 5 7 0 0 0 8 0 00 0 A87 0 0 0 0 0 0 0 0  487      R(INTR)=T, *LINK INTR.LIMIT.2.11;                       2573.000
0442 0 5 7 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                                  2574.000
0443 0 0 0 3 1 E 1 0 02 0 010 0 0 0 0 0 0 0 0           S=@00000001;    ENABLE INTERRUPT REQUEST                2575.000
0444 6 0 0 0 4 A B 3 02 0 010 0 0 0 0 0 0 0 0           FR(INTRTAB)=@01000000:FR(INTRTAB); SET RI PENDING BIT   2576.000
                                                    EI.EXIT                                                     2577.000
0445 0 6 0 0 1 E F 0 00 0 35D 0 0 0 0 0 0 0 0  35D      T=S, *GO TO DO.INTR.CTL;                                2578.000
                                                                                                                2579.000
                                                    **                                                          2580.000
                                                    **          DISABLE INTERRUPT INSTRUCTION                   2581.000
                                                    **                                                          2582.000
                                                    DI.INST                                                     2583.000
0446 0 5 7 0 0 0 8 0 00 0 A87 0 0 0 0 0 0 0 0  487      R(INTR)=T, *LINK INTR.LIMIT.2.11;                       2584.000
0447 0 5 7 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                                  2585.000
0448 0 0 0 3 1 E 1 0 02 0 020 0 0 0 0 0 0 0 0           S=@00000002;    DISABLE INTERRUPT REQUEST               2586.000
0449 E 4 0 4 4 D B 3 02 0 BC5 0 0 0 0 0 0 0 0  445      FR(INTRTAB)=@BCFFFFFF&FR(INTRTAB), TF %NCTR4 *HOP EI.EXIT;  2587.000
```

```
                SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                F1,DI,RI,DAI,AI
PC   TSMAB+DR YX PCH+BDM+BAD  ADDR

                                                                                                    2588.000
                                                    **                                              2589.000
                                                    **          REQUEST INTERRUPT INSTRUCTION       2590.000
                                                    **                                              2591.000
                                                    RI                                              2592.000
044A 0 5 7 0 0 0 8 0 00 0 A87 0 0 0 0 0 0 0 0  487      R(INTR)=T, *LINK INTR.LIMIT.2.11;           2593.000
044B 0 5 7 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                      2594.000
                                                    REQ.INTR                                        2595.000
044C 0 0 0 3 1 E 1 0 02 0 100 0 0 0 0 0 0 0 0           S=@00000010;    REQUEST INTERRUPT REQUEST   2596.000
044D 6 4 0 0 4 A B 3 02 0 02F 0 0 0 0 0 0 0 0  44E      FR(INTRTAB)=@02000000:FR(INTRTAB),*HOP RI.EXIT;  2597.000
                                                    RI.EXIT                                         2598.000
044E 0 6 0 0 1 E F 0 00 0 35D 0 0 0 0 0 0 0 0  35D      T=S, *GO TO DO.INTR.CTL;                    2599.000
                                                                                                    2600.000
                                                    **                                              2601.000
                                                    **          ACTIVATE INTERRUPT INSTRUCTION      2602.000
                                                    **                                              2603.000
                                                    AI                                              2604.000
044F 0 5 7 0 0 0 8 0 00 0 A87 0 0 0 0 0 0 0 0  487      R(INTR)=T, *LINK INTR.LIMIT.2.11;           2605.000
0450 0 5 7 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                      2606.000
0451 0 0 0 3 1 E 1 0 02 0 040 0 0 0 0 0 0 0 0           S=@00000004;    ACTIVATE INTERRUPT REQUEST  2607.000
0452 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0           NOD=@04000000&FR(OFFSET);       TEST IF ALREADY ACTIVE  2608.000
0453 6 0 0 0 4 A B 3 02 0 040 0 0 0 0 0 0 0 0           FR(INTRTAB)=@04000000:FR(INTRTAB);          2609.000
0454 B 6 0 0 1 E F 0 00 0 35D 0 0 0 0 0 0 0 0  35D      T=S,IF ALUZ *GO TO DO.INTR.CTL;             2610.000
                                                    AI.EXIT                                         2611.000
0455 0 5 0 0 1 0 0 0 00 0 08D 0 0 0 0 0 0 0 0  48D      *GO TO INTR.CTL.EXIT;                       2612.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          E I , D I , R J , D A I , A I
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2613.000
                                                    **                                                              2614.000
                                                    **          DEACTIVATE INTERRUPT INSTRUCTION                    2615.000
                                                    **                                                              2616.000
                                                    DAI                                                             2617.000
0456 0 5 7 0 0 0 8 0 00 0 A87 0 0 0 0 0 0 0 0  487          R(INTR)=T, *LINK INTR.LIMIT.2.11;                      2618.000
0457 0 5 7 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  467          *LINK BUILD.INTR.ADDR;                                 2619.000
0458 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0                NOD=@04000000&FR(OFFSET);       TEST IF LEVEL ACTIVE   2620.000
0459 6 0 0 0 4 B B 3 02 0 040 0 0 0 0 0 0 0 0                FR(INTRTAB)=%@04000000&FR(INTRTAB);                    2621.000
045A B 4 0 3 1 E F 0 02 6 085 0 0 0 0 0 0 0 0  455          T=@00000008, SDEST, IF ALUZ *GO TO AI.EXIT; SET DEACTIVATE BUS  2622.000
                                                    *                CODE AND EXIT IF LEVEL IS ALREADY ACTIVE       2623.000
045B 4 5 7 0 1 0 0 0 00 C 065 0 0 0 0 0 0 0 0  465          IF MODE75S *LINK DAI.INTFF.FLAG;                       2624.000
045C 0 6 0 0 1 0 0 0 00 0 350 0 0 0 0 0 0 0 0  350          *GO TO DO.INTR.CTL;                                    2625.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                          E X I T S - L A T E R R / E X T L
PC   T S M A B + D R  Y X PCH + B D M + R A D  ADDR
                                                                                                                2626.000
                                                                EXTPROC1                                        2627.000
045D 0 6 7 0 1 0 0 0 00 0 26D 0 0 0 0 0 0 0 0  26D      *LINK RESET.HIREG.RTN;                                  2628.000
045E 6 0 0 3 4 A B 7 02 0 020 0 0 0 0 0 0 0 0           FR(TRACE)=@00000002:FR(TRACF);  FLAG ENTRY FROM EXTL    2629.000
045F 0 6 0 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0    3      *GO TO EXTPROC;                                         2630.000
                                                                                                                2631.000
                                                                LEXTPROC                                        2632.000
0460 0 6 7 0 1 0 0 0 00 0 26D 0 0 0 0 0 0 0 0  26D      *LINK RESET.HIREG.RTN;                                  2633.000
0461 6 0 0 3 4 A B 7 02 0 040 0 0 0 0 0 0 0 0           FR(TRACE)=@00000004:FR(TRACE);  FLAG ENTRY FROM LATERR  2634.000
0462 1 4 0 0 1 0 0 0 00 0 EF4 0 0 0 0 0 0 0 0  464      IF ADDRSTOP *HOP S+2;                                   2635.000
0463 0 6 0 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0    3      *GO TO EXTPROC;                                         2636.000
0464 0 6 0 0 1 E E 0 02 0 0D8 0 0 0 0 0 0 0 0   D8      FULLMAR=(ADDR.STOP↑20R@FF000000), *GO TO ADDR.STOP;     2637.000
                                                                DAT.INTFF.FLAG                                  2638.000
0465 6 0 0 2 4 A B 4 02 0 800 0 0 0 0 0 0 0 0           FR(OFFSET)=@00008000:FR(OFFSET); SET INTFF OFF FLAG     2639.000
0466 0 1 0 0 1 E F 0 00 0 704 0 0 0 0 0 0 0 0           T=S, RESET(ENAINTFF), *JUMPJ;  NON INTERRUPTABLE IN 75 MODE  2640.000
```

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2641.000

                                                                                                                    2642.000
                                    **                                                                              2643.000
                                    ** BUILD INTERRUPT ADDRESS                                                     2644.000
                                    **     ENTRY PARAMETERS :                                                      2645.000
                                    **       T = INTERRUPT LEVEL IN BITS 9-15                                      2646.000
                                    **                                                                            2647.000
                                    **     EXIT PARAMETERS :                                                       2648.000
                                    **       R(RDEV) = SCRATCH PAD DEVICE ENTRY (IF NOT RTOM INTERRUPT)           2649.000
                                    **       R(INTR) = CHANNEL OR RTOM INTERRUPT LEVEL IN BITS 9-15               2650.000
                                    **       R(INTRLOC) = SCRATCH PAD INTERRUPT ENTRY ADDR                        2651.000
                                    **       R(INTRTAB) = SCRATCH PAD INTERRUPT ENTRY                             2652.000
                                    **       R(OFFSET) = SCRATCH PAD INTERRUPT ENTRY                              2653.000
                                    **       R(SI.VECTOR) = SI VECTOR  (CONTENTS OF SI VECTOR ADDR)               2654.000
                                    **       R(N.PSW2) = NEW PSW WORD 2  (75 MODE ONLY)                           2655.000
                                    **                                                                            2656.000

                                                                                                                    2657.000

                                    (@467)                                                                          2658.000

                                    BUILD.INTR.ADDR                                                                 2659.000

0467 6 0 0 1 0 A B 5 02 6 800 0 0 0 0 0 0 0 0       FR(INTRLOC)=@00800000:T,SDEST; MAKE SCRATCH PAD INT ENTRY ADDR  2660.000

0468 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0       S=SCRATCH(S);          READ INTR TABLE ENTRY                    2661.000

0469 0 0 0 0 1 E 8 0 00 0 800 0 0 0 0 0 0 0 0       R(INTRTAB)=S;                                                   2662.000

046A 0 0 0 0 0 0 E 0 00 F 00D 0 0 0 0 0 0 0 0       FULLMAR=T(ZE), CLDNU;  LOAD SI VECTOR ADDR INTO MAR             2663.000

046B 6 0 0 6 0 D B 4 02 0 000 0 0 0 0 0 0 0 0       FR(OFFSET)=@FFFF00FF&T; CLEAR BITS 16-23 OF INTERRUPT ENTRY     2664.000

046C 0 0 0 0 1 E F 0 1C 1 080 0 0 0 0 0 0 0 0       T=S, READ, FRCWORD; FETCH ST VECTOR FROM MEMORY                 2665.000

046D 0 0 0 1 0 D 0 0 02 0 800 0 0 0 0 0 0 0 0       NOD=@00800000&T;     TEST FOR RTOM                              2666.000

046E 6 0 0 1 0 D B 0 02 6 7F0 0 0 0 0 0 0 0 0       FR(EDEV)=@007F0000&T,SDEST;       CLEAR UNUSED BITS             2667.000

046F B 4 0 5 0 B 1 0 02 2 F0A 0 0 0 0 0 0 0 0  47A  S=X@FFF0FFFF&T, TNIBL, IF ALUZ *GO TO INTR.IOM.TYPE;            2668.000

                                    *                    ;GET RTOM SUB-ADDR IN S REG BITS 12-15 AND                 2669.000

                                    *                    ;SHIFT RTOM PHYSICAL ADDR TO T REG BITS 4-7                2670.000

                                    INTR.RTOM.TYPE                                                                  2671.000

0470 C 4 0 0 0 D F 0 02 0 0F2 0 0 0 0 0 0 0 0  472  T=@0F000000&T, IF NALUZ *GO TO $+2; GET LS3-BITS OF RTOM ADDR    2672.000

0471 0 5 0 0 1 0 0 0 00 0 085 0 0 0 0 0 0 0 0  485  *GO TO INVALID.IOM; INVALID RTOM PHYSICAL ADDRESS OF ZERO        2673.000

0472 0 0 0 0 0 A F 0 02 0 780 0 0 0 0 0 0 0 0       T=@78000000:T;          MSS-BITS OF RTOM BOARD ADDR             2674.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INTERRUPT SUB-ROUTINES

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0473 0 0 0 0 0 A 8 0 00 0 900 0 0 0 0 0 0 0 0          R(RDEV)=S:T;            COMBINE RTOM BOARD AND SUB ADDR          2675.000
0474 0 0 0 1 1 E 1 0 02 0 7F0 0 0 0 0 0 0 0 0          S=@007F0000;                                                     2676.000
0475 0 0 0 0 5 0 F 0 00 F 900 0 0 0 0 0 0 0 0          T=R(RDEV,HWS,7E);                                                2677.000
0476 0 0 0 0 4 C 1 0 00 0 A00 0 0 0 0 0 0 0 0          S=S&%R(INTR); COMPLEMENT INTERRUPT LEVEL                         2678.000
0477 0 4 0 0 0 A 1 0 00 0 00B 0 0 0 0 0 0 0 0  47B     S=S:T, *GO TO SAVE.DVC.ENTRY; COMBINE INTR LVL WITH RTOM ADDR    2679.000
                                                BUILD.INTR.EXIT                                                     2680.000
0478 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0          IF %OPNORESP:OPTIMEOUT:OPRNDPE *JUMPJ; TEST FOR MEM ERRORS       2681.000
0479 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530     NU=T(7E), *GO TO CURRENT.INST.ERROR;                             2682.000
                                                INTR.IOM.TYPE                                                       2683.000
047A 0 0 0 4 1 0 1 0 00 A 800 0 0 0 0 0 0 0 0          S=SCRATCH(R(EDEV));          READ DEVICE ENTRY                   2684.000
                                                SAVE.DVC.ENTRY                                                      2685.000
0478 0 5 7 0 1 E 8 0 00 0 986 0 0 0 0 0 0 0 0  486     R(RDEV)=S, *LINK DUD.INTR;   SAVE DEVICE ENTRY                   2686.000
                                                RTOM.IOM.EXIT                                                       2687.000
047C E 5 0 4 3 D F 0 00 D 382 0 0 0 0 0 0 0 0  482     T=FR(PCMASK)&DI, OTHERBANK, IF %MODE75 *GO TO N.75.EXIT;         2688.000
047D 0 0 0 3 0 3 E 0 02 0 0C0 0 0 0 0 0 0 0 0          FULLMAR=@0000000C+T; COMPUTE ADDRESS OF N.PSW2                   2689.000
047E 0 4 7 0 0 0 8 0 1C 1 688 0 0 0 0 0 0 0 0  478     R(ST.VECTOR)=T,READ,FRCWORD,*LINK BUILD.INTR.EXIT;FETCH N.PSW2   2690.000
047F 0 4 7 0 1 E 0 0 00 0 006 0 0 0 0 0 0 0 0  486     NOD=S, *LINK DUD.INTR;  TEST FOR ZERO DVC ENTRY                  2691.000
0480 8 4 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0  485     IF ALUZ *GO TO INVALID.IOM;  EXIT IF DVC ENTRY =0                2692.000
0481 0 5 0 0 3 0 8 0 00 0 078 0 0 0 0 0 0 0 0  476     R(N.PSW2)=DI, *GO TO BUILD.INTR.EXIT; SAVE WORD 2 OF NEW PSW     2693.000
                                                N.75.EXIT                                                           2694.000
0482 0 0 0 0 1 E 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=S; TEST FOR ZERO DVC ENTRY                                   2695.000
0483 0 0 0 0 0 0 8 0 00 0 600 0 0 0 0 0 0 0 0          R(ST.VECTOR)=T;  SAVE SI VECTOR                                  2696.000
0484 C 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          IF NALUZ *JUMPJ; RETURN IF DVC ENTRY NOT =0                      2697.000
                                                INVALID.IOM                                                         2698.000
0485 6 0 0 4 1 E B 1 02 0 000 0 0 0 0 0 0 0 0          FR(RDEV)=@00FFFFFF; GENERATE A BAD DEVICE ENTRY                  2699.000
                                                DUD.INTR                                                            2700.000
0486 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *JUMPJ;                                                          2701.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  79APR27
SFL  32/75  CPU                                                          I N T E R R U P T   S U B - R O U T I N E S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    2702.000

                                                                                                                    2703.000
                                                    **                                                              2704.000
                                                    **          EXCLUDE INTERRUPT LEVELS 02 THRU 11                 2705.000
                                                    **                                                              2706.000

                                                                                                                    2707.000

                                                    INTR.LIMIT.2.11                                                 2708.000

0487 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0            S=T; SAVE  INTERRUPT LEVEL                                 2709.000

0488 0 0 0 0 5 0 F 0 00 F A0D 0 0 0 0 0 0 0 0            T=R(INTR,HWS,7E), CLDNU;                                   2710.000

0489 0 0 0 7 0 3 0 0 00 0 FE0 0 0 0 0 0 0 0 0            NOD=@FFFFFFFE+T; TEST LEVEL < 2                            2711.000

048A 0 0 0 7 0 3 0 0 00 0 FE0 0 0 0 0 0 0 0 0            NOD=@FFFFFFEE+T; TEST LEVEL > 11                           2712.000

048B 3 1 0 0 4 0 F 0 00 2 A00 0 0 0 0 0 0 0 0            T=R(INTR), IF ALUNEGW *JUMPJ; EXIT OK, LEVEL < 2           2713.000

048C A 1 0 0 1 0 0 0 00 0 800 0 0 0 0 0 0 0 0            IF %ALUNEG *JUMPJ; EXIT OK LEVEL > @11                     2714.000

                                                    INTR.CTL.EXIT                                                   2715.000

048D 0 3 0 0 1 0 0 0 19 0 084 0 0 0 0 0 0 0 0            FETCHPC,*JUMPZ,RESET(HIREG);                               2716.000

                                                    TD.CLASSF.ERROR                                                 2717.000

048E 4 6 0 0 2 0 0 0 00 C 3F0 0 0 0 0 0 0 0 0  3F0       NOD=IO, IF MODE75S *GO TO EXCLUDE.CD.CLASSF;                2718.000

048F 5 4 0 0 3 E 1 0 02 8 102 0 0 0 0 0 0 0 0  492       S=@10000000,BMUX=DI, IF %BMUX16 *GO TO TD.CLASSF.ERR.EXIT;  2719.000

0490 5 4 0 0 2 0 0 0 00 9 002 0 0 0 0 0 0 0 0  492       NOD=IO, IF %BMUX17 *GO TO TD.CLASSF.ERR.EXIT;               2720.000

0491 0 0 0 0 1 E 1 0 02 0 200 0 0 0 0 0 0 0 0            S=@20000000; SET TD2000 CC 2                               2721.000

                                                    TD.CLASSF.ERR.EXIT                                              2722.000

0492 0 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C       *GO TO TD2.WRITE.CC;   EXIT                                2723.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
SEL  32/75  CPU                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2724.000
                                                        TD                                                          2725.000
0493 0 6 7 0 1 0 0 0 00 0 2DB 0 0 0 0 0 0 0 0  2DB          *LINK IO.REG.LOAD;                                     2726.000
0494 6 0 0 0 4 8 0 1 02 0 0F0 0 0 0 0 0 0 0 0                NOD=@0F000000!FR(RDEV); TEST FOR CLASS F DEVICE        2727.000
0495 0 0 0 3 1 E F 0 02 0 080 0 0 0 0 0 0 0 0                T=@00000008; ARSTX/RSTX CODE FOR SA&CTRLR STATUS       2728.000
0496 2 5 0 0 1 0 0 0 00 0 78E 0 0 0 0 0 0 0 0  48E          IF ALU4-77 *GO TO TD.CLASSF.ERROR;                     2729.000
0497 0 6 7 0 1 0 0 0 00 0 323 0 0 0 0 0 0 0 0  323          *LINK CMD.IOC;  DO ARSTX/RSTX SEQ                      2730.000
0498 0 5 7 0 1 0 0 0 00 0 0BB 0 0 0 0 0 0 0 0  4BB          *LINK TD.TEST.STATUS;                                  2731.000
0499 6 0 0 0 4 8 0 1 02 0 0E0 0 0 0 0 0 0 0 0                NOD=@0E000000!FR(RDEV); TEST FOR CLASS E               2732.000
049A 0 0 0 0 3 0 8 0 00 0 100 0 0 0 0 0 0 0 0                R(TD84)=DI; SAVE 8000/4000 STATUS                      2733.000
049B A 4 0 0 2 0 0 0 00 0 70D 0 0 0 0 0 0 0 0  49D          NOD=IO, IF %ALU4-7Z *GO TO TD.2000 ;                   2734.000
049C 3 5 0 0 2 0 0 0 00 A 0E6 0 0 0 0 0 0 0 0  4F6          NOD=IO, IF BMUX18 *GO TO TD.DEV.CLASSE;                2735.000
                                                        TD.2000                                                     2736.000
049D 0 0 0 3 1 E F 0 02 0 040 0 0 0 0 0 0 0 0                T=@00000004;   PREPARE FOR DEV STATUS                   2737.000
049E 0 6 7 0 1 0 0 0 00 0 323 0 0 0 0 0 0 0 0  323          *LINK CMD.IOC;  SEND 2000 ARSTX/RSTX                   2738.000
049F 0 5 7 0 1 0 0 0 00 0 0BB 0 0 0 0 0 0 0 0  4BB          *LINK TD.TEST.STATUS;                                  2739.000
04A0 0 5 7 0 3 0 8 0 00 0 286 0 0 0 0 0 0 0 0  486          R(TD2)=DI, *LINK DUD.INTR;    SAVE 2000 LEVEL STATUS   2740.000
04A1 6 0 0 0 4 8 0 1 02 0 0E0 0 0 0 0 0 0 0 0                NOD=@0E000000!FR(RDEV);  TEST FOR CLASS E              2741.000
04A2 6 0 0 0 4 8 0 1 02 0 027 0 0 0 0 0 0 0 0                NOD=@02000000!FR(RDEV), CLRS; TEST FOR CLASS 2         2742.000
04A3 2 5 0 0 2 0 0 0 00 0 7E6 0 0 0 0 0 0 0 0  4E6          NOD=IO, IF ALU4-77 *GO TO TD.DEV.CLASSE;               2743.000
04A4 2 5 0 0 2 0 0 0 00 0 7D9 0 0 0 0 0 0 0 0  4D9          NOD=IO, IF ALU4-77 *GO TO TD.DEV.CLASS2;               2744.000
04A5 6 0 0 0 4 8 0 1 02 0 010 0 0 0 0 0 0 0 0                NOD=@01000000!FR(RDEV); TEST FOR CLASS 1               2745.000
04A6 6 0 0 0 4 8 0 1 02 0 030 0 0 0 0 0 0 0 0                NOD=@03000000!FR(RDEV); TEST FOR CLASS 3               2746.000
04A7 2 5 0 0 2 0 0 0 00 0 7C7 0 0 0 0 0 0 0 0  4C7          NOD=IO, IF ALU4-77 *GO TO TD.DEV.CLASS1;               2747.000
04A8 2 5 0 0 2 0 0 0 00 0 7E6 0 0 0 0 0 0 0 0  4E6          NOD=IO, IF ALU4-77 *GO TO TD.DEV.CLASSE; CLASS 3 EXIT  2748.000
                                                                                                                    2749.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                2750.000
                                                    *                                                                           2751.000
                                                    *          8600 LINE PRINTER STATUS FORMATTON                               2752.000
                                                    *                                                                           2753.000
                                                    TD.DEV.CLASS0                                                               2754.000
04A9 5 5 0 0 2 0 0 0 00 8 0C1 0 0 0 0 0 0 0 0  4C1          NOD=I0, IF %BMUX16 *GO TO LP.TD8;  TD 8000?                     2755.000
04AA 5 5 0 0 1 0 0 0 00 9 08F 0 0 0 0 0 0 0 0  48F          IF %BMUX17 *GO TO LP.TD4;  TD 4000?                             2756.000
                                                    *                                                                           2757.000
                                                    *                                                                           2758.000
                                                    LP.TD2                                                                      2759.000
                                                    *                                                                           2760.000
                                                    *              LP TD2000 STATUS REQUESTED                                   2761.000
                                                    *                                                                           2762.000
04AB 0 0 0 0 4 0 0 0 00 0 107 0 0 0 0 0 0 0 0          NOD=R(TD84),CLRS;  CHECK FOR DEV PROG VIOL (INV CMD)                    2763.000
04AC 0 0 0 0 1 E E 0 02 6 700 0 0 0 0 0 0 0 0          FULLMAR=@70000000,SDEST;                                                2764.000
04AD A 4 0 0 1 E F 0 00 0 80F 0 0 0 0 0 0 0 0  4AF      IF %ALUNEG *GO TO $+2, T=S;  IF NOT PROG VIOL                           2765.000
04AE 0 0 0 2 1 E E 0 02 0 40D 0 0 0 0 0 0 0 0          FULLMAR=@00004000, CLDNU;  SET PRO VIOL (BIT 1/17)                      2766.000
04AF 0 0 0 4 0 D F 0 00 0 207 0 0 0 0 0 0 0 0          T=R(TD2) & T,CLRS;                                                      2767.000
04B0 0 0 0 0 0 D D 0 02 0 700 0 0 0 0 0 0 0 0          NL=@70000000 & T; TEST FOR INOP CONDITIONS                              2768.000
04B1 0 0 0 0 1 E F 0 02 0 080 0 0 0 0 0 0 0 0          T=@08000000;                                                            2769.000
04B2 B 4 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  4B4      IF ALUZ *GO TO $+2;  IF NO INOP COND                                    2770.000
04B3 0 0 0 2 1 A E 0 02 0 20D 0 0 0 0 0 0 0 0          FULLMAR=@00002000 : MAR, CLDNU;  SET INOP (BIT 2/18)                    2771.000
04B4 0 5 7 4 0 D 0 0 00 0 286 0 0 0 0 0 0 0 0  4B6      NOD=R(TD2) & T,*LINK DUD.INTR;                                          2772.000
04B5 B 4 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0  4B7      IF ALUZ *GO TO $+2;  IF NOT BOF                                         2773.000
04B6 0 0 0 3 1 A E 0 02 0 40D 0 0 0 0 0 0 0 0          FULLMAR=@00000040 : MAR, CLDNU;  SET BOF (BIT 9/25)                     2774.000
04B7 0 0 0 0 4 0 0 0 00 0 200 0 0 0 0 0 0 0 0          NOD=R(TD2);                                                             2775.000
04B8 3 6 0 0 1 0 6 0 00 8 507 0 0 0 0 0 0 0 0  507      DI=MAR,IF BMUX16 *GO TO TD2.WRITE;  IF NOT BUSY                         2776.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

04B9 0 0 0 3 1 A E 0 02 0 04D 0 0 0 0 0 0 0 0          FULLMAR=@00000004 : MAR, CLDNU;  SET DEV BUSY (BIT 13/29)        2777.000

04BA 0 6 0 0 1 0 6 0 00 0 507 0 0 0 0 0 0 0 0  507     DI=MAR,*GO TO TD2.WRITE;                                         2778.000

                                                TD.TEST.STATUS                                                          2779.000

04BB 2 1 0 0 1 0 0 0 00 0 900 0 0 0 0 0 0 0 0          IF NCTRZ *JUMPJ;                                                 2780.000

04BC 9 4 0 0 1 0 0 0 00 0 BBE 0 0 0 0 0 0 0 0  4BE     IF %IOCHBUSY *GO TO $+2;                                         2781.000

04BD 0 1 0 3 1 E 6 0 02 0 040 0 0 0 0 0 0 0 0          DI=@00000004,*JUMPJ;     SET ACTIVE                              2782.000

                                                                                                                        2783.000
                                                ** 
                                                **              IOC BROKEN OR NOT PRESENT                               2784.000
                                                **                                                                      2785.000
                                                **                                                                      2786.000

                                                                                                                        2787.000

                                                TD.NO.IOC                                                               2788.000

04BE 0 6 0 0 0 1 1 0 00 F 50C 0 0 0 0 0 0 0 0  50C     S=%T(ZE),*GO TO TD2.WRITE.CC;                                    2789.000

                                                                                                                        2790.000
                                                *                                                                       2791.000
                                                *       LINE PRINTER TD4000 STATUS REQUEST                              2792.000
                                                *                                                                       2793.000

                                                                                                                        2794.000

                                                LP.TD4                                                                  2795.000

04BF 0 5 7 0 1 0 0 0 00 0 0F6 0 0 0 0 0 0 0 0  4F6     *LINK TD4.DEV;                                                   2796.000

04C0 0 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C     *GO TO TD2.WRITE.CC;                                             2797.000

                                                                                                                        2798.000
                                                *                                                                       2799.000
                                                *       LINE PRINTER TD8000 STATUS REQUEST                              2800.000
                                                *                                                                       2801.000

                                                                                                                        2802.000

                                                LP.TD8                                                                  2803.000

04C1 0 5 7 0 1 0 0 0 00 0 0FC 0 0 0 0 0 0 0 0  4FC     *LINK TD8.DEV;                                                   2804.000

04C2 0 0 0 0 0 0 0 0 02 0 780 0 0 0 0 0 0 0 0          NOD=@78000000 & T;  TEST FOR TD2000 STATUS                       2805.000

04C3 0 0 0 0 1 0 1 0 00 0 000 0 0 0 0 0 0 0 0          S=MAR;                                                           2806.000

04C4 8 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  4C6     IF ALUZ *GO TO $+2;  IF NO DEV STATUS                            2807.000

04C5 0 0 0 0 1 A 1 0 02 0 080 0 0 0 0 0 0 0 0          S=@08000000 : MAR;  SET DEV STAT CC'S (CC4)                      2808.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                                    T D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

04C6 0 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C        *GO TO TD2.WRITE.CC;                    2809.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    TD
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    2810.000
                                                *                                                                   2811.000
                                                *       8600 CARD READER STATUS FORMATION                           2812.000
                                                *                                                                   2813.000
                                                TD.DEV.CLASS1                                                       2814.000
04C7 5 4 0 0 2 0 0 0 00 8 00F 0 0 0 0 0 0 0 0  4CF      NOD=I0, IF %BMUX16 *GO TO CR.TD8;  TD8000?                  2815.000
04C8 5 6 0 0 1 0 0 0 00 A 50C 0 0 0 0 0 0 0 0  50C      IF %BMUX18 *GO TO TD2.WRITE.CC;  TD2000?                    2816.000

                                                                                                                    2817.000
                                                *                                                                   2818.000
                                                CR.TD4                                                              2819.000
                                                *                                                                   2820.000
                                                *       CARD RDR TD4000 STATUS REQ                                  2821.000
                                                *                                                                   2822.000
04C9 0 5 7 0 1 0 0 0 00 0 0F6 0 0 0 0 0 0 0 0  4F6      *LINK TD4.DEV;                                              2823.000
04CA 0 0 0 0 1 E F 0 02 0 040 0 0 0 0 0 0 0 0          T=@04000000;                                                2824.000
04CB 0 0 0 4 0 D 0 0 00 0 200 0 0 0 0 0 0 0 0          NOD=R(TD2) & T;                                             2825.000
04CC 0 0 0 0 1 E F 0 02 0 080 0 0 0 0 0 0 0 0          T=@08000000;                                                2826.000
04CD B 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C      IF ALUZ *GO TO TD2.WRITE.CC;  IF NOT UNDERFLOW (DATALOST)   2827.000
04CE 0 6 0 0 0 A 1 0 00 0 50C 0 0 0 0 0 0 0 0  50C      S=S:T, *GO TO TD2.WRITE.CC; SET CC4                         2828.000
                                                                                                                    2829.000
                                                *                                                                   2830.000
                                                *                                                                   2831.000
                                                CR.TD8                                                              2832.000
04CF 0 5 7 0 1 0 0 0 00 0 0FC 0 0 0 0 0 0 0 0  4FC      *LINK TD8.DEV;                                              2833.000
04D0 0 0 0 0 0 D 0 0 02 0 F00 0 0 0 0 0 0 0 0          NOD=@F0000000 & T; TEST FOR INOP COND'S                     2834.000
04D1 0 0 0 0 1 0 1 0 00 0 000 0 0 0 0 0 0 0 0          S=MAR;                                                      2835.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

04D2 B 4 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  4D4    IF ALUZ *GO TO $+2;                          2836.000

04D3 0 0 0 0 1 A E 0 02 0 08D 0 0 0 0 0 0 0 0         FULLMAR=@08000000:MAR,CLDNU;                 2837.000

04D4 0 0 0 0 0 D 0 0 02 0 040 0 0 0 0 0 0 0 0         NOD=@04000000&T;                             2838.000

04D5 0 0 0 0 1 0 1 0 00 0 000 0 0 0 0 0 0 0 0         S=MAR;                                       2839.000

04D6 B 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C    IF ALUZ *GO TO TD2.WRITE.CC;                 2840.000

04D7 0 0 0 0 1 A E 0 02 0 10D 0 0 0 0 0 0 0 0         FULLMAR=@10000000:MAR,CLDNU;                 2841.000

04D8 0 6 0 0 1 0 1 0 00 0 50C 0 0 0 0 0 0 0 0  50C    S=MAR,*GO TO TD2.WRITE.CC;                   2842.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU TD

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                2843.000
                                                        *                                                       2844.000
                                                        *                                                       2845.000
                                                        *               8600 TTY STATUS FORMATION               2846.000
                                                        *                                                       2847.000
                                                        *                                                       2848.000
                                                        TD.DEV.CLASS2                                           2849.000
04D9 5 5 0 0 2 0 0 0 00 8 0E1 0 0 0 0 0 0 0 0  4F1              NOD=IO, IF %BMUX16 *GO TO TY.TD8;  TD8000?      2850.000
04DA 5 6 0 0 1 0 0 0 00 A 50C 0 0 0 0 0 0 0 0  50C              IF %BMUX18 *GO TO TD2.WRITE.CC; TD2000?         2851.000
                                                        *                                                       2852.000
                                                                                                                2853.000
                                                        *                                                       2854.000
                                                        *               8600 TTY TD4000 STATUS                  2855.000
                                                        *                                                       2856.000
                                                        TY.TD4                                                  2857.000
04DB 0 5 7 0 1 0 0 0 00 0 0F6 0 0 0 0 0 0 0 0  4F6              *LINK TD4.DEV;                                  2858.000
04DC 0 0 0 0 1 E F 0 02 0 600 0 0 0 0 0 0 0 0                   T=@60000000;                                    2859.000
04DD 0 0 0 4 0 D 0 0 00 0 200 0 0 0 0 0 0 0 0                   NOD=R(TD2) & T;                                 2860.000
04DE 0 0 0 0 1 E F 0 02 0 080 0 0 0 0 0 0 0 0                   T=@08000000;                                    2861.000
04DF 8 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C              IF ALU7 *GO TO TD2.WRITE.CC;                    2862.000
04E0 0 6 0 0 0 A 1 0 00 0 50C 0 0 0 0 0 0 0 0  50C              S=S:T, *GO TO TD2.WRITE.CC; SET CC4             2863.000
                                                                                                                2864.000
                                                        *                                                       2865.000
                                                        *               8600 TTY TD8000                         2866.000
                                                        *                                                       2867.000
                                                        TY.TD8                                                  2868.000
04E1 0 5 7 0 1 0 0 0 00 0 0FC 0 0 0 0 0 0 0 0  4FC              *LINK TD8.DEV;                                  2869.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
SEL 32/75 CPU                                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

04F2 0 0 0 0 1 E F 0 02 0 080 0 0 0 0 0 0 0 0          T=@08000000;   SET CC4 IF TTY INOP (READY)                    2870.000

04E3 0 0 0 4 0 D 1 0 00 0 200 0 0 0 0 0 0 0 0          S=R(TD2) & T;                                                2871.000

04F4 0 6 0 0 1 A 1 0 00 0 50C 0 0 0 0 0 0 0 0  50C     S=S : MAR, *GO TO TD2.WRITE.CC; SET CC4 IF NOT RDY           2872.000

04F5 0 6 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0  50C     *GO TO TD2.WRITE.CC;                                         2873.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 79APR27
SEL 32/75 CPU TD
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    2874.000
                                                    *                                                               2875.000
                                                    *               8600 DEV CLASS 'E' STATUS                       2876.000
                                                    *                                                               2877.000
                                                    *                                                               2878.000
                                                    TD.DEV.CLASSE                                                   2879.000
04F6 5 5 0 0 2 0 0 0 00 8 0F1 0 0 0 0 0 0 0 0  4F1          NOD=I0, IF %BMUX16 *GO TO E.TD8;  TD8000?               2880.000
04F7 5 4 0 0 4 0 0 0 00 9 90C 0 0 0 0 0 0 0 0  4FC          NOD=R(RDEV),IF %BMUX17 *GO TO E.TD4; TD4000?            2881.000
                                                    *                                                               2882.000
                                                    *                                                               2883.000
                                                    *                                                               2884.000
                                                    E.TD2                                                           2885.000
04E8 0 0 0 0 5 0 6 0 00 6 200 0 0 0 0 0 0 0 0           DI=R(TD2,HWS),SDEST; SWAP TD2000 STAT AND CC'S              2886.000
04E9 0 0 0 2 1 E D 0 00 0 000 0 0 0 0 0 0 0 0           NL=SNIBL; POSITION CC'S TO 0-4                              2887.000
04EA 0 0 0 0 1 D 1 0 02 0 F00 0 0 0 0 0 0 0 0           S=@F0000000&N;    CLEAR ALL BUT CC'S                        2888.000
                                                    E.TD2.S                                                         2889.000
04EB 0 6 2 0 3 0 6 0 01 F 507 0 0 0 0 0 0 0 0  507      DI=DI(ZE), SHIFTS(#), *GO TO TD2.WRITE; SHIFT CC'S FOR SREG  2890.000
                                                                                                                    2891.000
                                                    *                                                               2892.000
                                                    *       'E' CLASS TD4000                                        2893.000
                                                    *                                                               2894.000
                                                    *                                                               2895.000
                                                    E.TD4                                                           2896.000
04EC 0 4 7 0 4 0 F 0 00 0 10F 0 0 0 0 0 0 0 0  4EF          T=R(TD84),*LINK TD4.ECLASS;                             2897.000
04ED 0 0 0 4 3 A 1 0 00 0 100 0 0 0 0 0 0 0 0           S=R(TD84):DI;           FOLD IN TD4000 CC'S             2898.000
                                                    E.TD4.S                                                         2899.000
04EE 0 6 2 0 1 0 0 0 01 0 50C 0 0 0 0 0 0 0 0  50C      SHIFTS(#), *GO TO TD2.WRITE.CC;                             2900.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    TD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                          **                                                    2901.000
                                          **          E-CLASS TD 4000/8000 STATUS               2902.000
                                          **                                                    2903.000
                                          TD4.ECLASS                                            2904.000
04FF 5 1 0 0 0 D 6 0 02 2 600 0 0 0 0 0 0 0 0         DI=@60000000&T,IF %ALUNEGW *JUMPJ;         2905.000
04F0 0 1 0 0 3 A 6 0 02 0 200 0 0 0 0 0 0 0 0         DI=@20000000:DI, *JUMPJ; SET PROGRAM VIOLATION FLAG  2906.000
                                                                                                2907.000
                                          *                                                     2908.000
                                          *                                                     2909.000
                                          *          'E' CLASS TD8000                           2910.000
                                          *                                                     2911.000
                                          *                                                     2912.000
                                          F.TD8                                                 2913.000
04F1 0 0 0 0 4 0 0 0 00 0 900 0 0 0 0 0 0 0 0         NOD=R(RDEV);          TEST FOR CP PROG VIOL  2914.000
04F2 0 5 7 0 4 0 F 0 00 0 1EF 0 0 0 0 0 0 0 0 4FF     T=R(TD84),*LINK TD4.ECLASS;                 2915.000
04F3 0 0 0 0 4 0 1 0 00 0 100 0 0 0 0 0 0 0 0         S=R(TD84);            RELOAD 4000/8000 IOC GEN STATUS  2916.000
04F4 0 0 0 0 3 0 1 0 00 2 080 0 0 0 0 0 0 0 0         S=DT,TNIBR;           ISOLATE 8000 STATUS IN T  2917.000
04F5 0 5 0 0 0 A 1 0 00 0 0EE 0 0 0 0 0 0 0 0 4FE     S=S:T,*GO TO F.TD4.S;                       2918.000
                                                                                                2919.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    TD
PC  TSMAB+DR YX PCH+BDM+BAD ADDR
                                                                                                                    2920.000
                                                *                                                                   2921.000
                                                *                                                                   2922.000
                                                *       COMMON NON D OR E-CLASS TD4000 STATUS LOGIC                 2923.000
                                                *                                                                   2924.000
                                                TD4.DEV                                                             2925.000
04F6 0 0 0 0 4 0 0 0 00 0 900 0 0 0 0 0 0 0 0           NOD=K(KDEV);                                                2926.000
04F7 0 0 0 0 1 E F 0 02 0 607 0 0 0 0 0 0 0 0           T=@60000000, CLRS;                                          2927.000
04F8 A 4 0 0 1 0 0 0 00 0 80A 0 0 0 0 0 0 0 0  4FA      IF %ALUNEG *G0 TO $+2;  IF NOT PROG VIOL (DCC)              2928.000
04F9 0 0 0 0 1 E 1 0 02 0 100 0 0 0 0 0 0 0 0           S=@10000000;  SET PROG VIOL (DCC) CC3                       2929.000
04FA 0 0 0 4 0 D F 0 00 0 100 0 0 0 0 0 0 0 0           T=R(TD84) & T;                                              2930.000
04FB 0 1 0 0 0 A 1 0 00 0 000 0 0 0 0 0 0 0 0           S=S:T, *JUMPJ;                                              2931.000
                                                *                                                                   2932.000
                                                *       COMMON NON D OR E-CLASS TD8000 STATUS LOGIC                 2933.000
                                                *                                                                   2934.000
                                                *                                                                   2935.000
                                                TD8.DEV                                                             2936.000
04FC 0 0 0 3 1 E E 0 02 0 00D 0 0 0 0 0 0 0 0           FULLMAR=0, CLDNU;  CLEAR MAR                                2937.000
04FD 0 0 0 0 4 0 0 0 00 0 900 0 0 0 0 0 0 0 0           NOD=R(KDEV);                                                2938.000
04FE 0 0 0 0 4 0 F 0 00 0 107 0 0 0 0 0 0 0 0           T=K(TD84), CLRS;                                            2939.000
04FF 3 0 0 0 1 A 0 0 02 2 10D 0 0 0 0 0 0 0 0           NOD=@10000000:N,IF ALUNEGW CLDNU;       SET PROG VIOL CC3   2940.000
0500 0 0 0 0 4 0 0 0 00 0 100 0 0 0 0 0 0 0 0           NOD=R(TD84);            TEST DEV PROG VIOL                  2941.000
0501 0 0 0 0 0 D 0 0 02 0 600 0 0 0 0 0 0 0 0           NOD=@60000000&T;        TEST IF ANY 4000 STATUS             2942.000
0502 3 0 0 0 1 A 0 0 02 2 08D 0 0 0 0 0 0 0 0           NOD=@08000000:N,IF ALUNEGW CLDNU;       SET 2000 STAT CC4   2943.000
0503 C 0 0 0 1 A 0 0 02 0 10D 0 0 0 0 0 0 0 0           NOD=@10000000:N,IF NALUZ CLDNU;         SET 4000 STAT CC3   2944.000
0504 0 0 0 0 4 0 F 0 00 0 200 0 0 0 0 0 0 0 0           T=R(TD2);               LOAD 2000 STATUS                    2945.000
0505 5 0 0 0 1 A 0 0 02 8 20D 0 0 0 0 0 0 0 0           NOD=@20000000:N,IF %BMUX16 CLDNU;       SET BUSY STAT CC2   2946.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SFL  32/75  CPU                                                    TD
PC   TSMAB+DR  YX PCH+BDM+BAD  ADDR

                                                                                                                    2947.000
                                                DUD.50X                                                             2948.000
0506 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0         *JUMPJ;                                                       2949.000
                                                                                                                    2950.000
                                                *                                                                   2951.000
                                                *           TD2000 COMMON TERMINATION                               2952.000
                                                *                                                                   2953.000
                                                TD2.WRITE                                                           2954.000
0507 0 0 0 0 3 A F 0 00 0 000 0 0 0 0 0 0 0 0         T=S:DI ; COMBINE CC'S & STATUS                                2955.000
0508 0 0 0 0 0 D E 0 02 0 200 0 0 0 0 0 0 0 0         FULLMAR=@20000000&T; TEST FOR STATUS XFER NOT PERFORMED (CC 2)  2956.000
0509 0 0 0 0 4 0 6 0 00 0 300 0 0 0 0 0 0 0 0         DT=R(TCW);  FETCH TD TCW                                      2957.000
050A C 4 0 5 3 D 3 0 02 0 07C 0 0 0 0 0 0 0 0  50C    MAR=@FF07FFFF&DI,IF NALUZ *GO TO TD2.WRITE.CC;ALLOW MAPPED TD  2958.000
050B 0 0 0 0 1 0 0 0 1F 0 080 0 0 0 0 0 0 0 0         WRITE;   WRITE 2000 STAT TO MEM                               2959.000
                                                TD2.WRITE.CC                                                        2960.000
050C 0 0 0 0 1 0 0 0 00 0 501 0 0 0 0 0 0 0 0         SETCC(S);   SET COND CODES FROM S REG                         2961.000
050D 0 6 0 0 1 0 0 0 00 0 3A4 0 0 0 0 0 0 0 0  3A4    *GO TO CD.EXIT;                                               2962.000
                                                                                                                    2963.000
                                                TEST.FULL.SCALE.NEGATIVE                                            2964.000
050E A 1 0 3 1 E 1 0 00 2 480 0 0 0 0 0 0 0 0         S=SNIBR,TNIBR, IF %SIGNSAVE7 *JUMPJ;                          2965.000
050F 0 0 0 3 1 E 1 0 00 2 080 0 0 0 0 0 0 0 0         S=SNIBR,TNIBR; SET UP TO HANDLE MAX FULL SCALE NEG            2966.000
0510 0 0 0 0 0 D C 0 02 0 FF0 0 0 0 0 0 0 0 0         NU=@FF000000&T;                                               2967.000
0511 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0         R(TMP4)=N;                                                    2968.000
0512 0 0 0 0 1 E F 0 00 0 F05 0 0 0 0 0 0 0 0         T=S, SET(FLAG); SET MAX FULL SCALE NEG FLAG                   2969.000
0513 0 0 0 3 0 3 1 0 02 0 009 0 0 0 0 0 0 0 0         S=0+T,ABST;                                                   2970.000
0514 0 6 0 0 0 1 C 0 00 F 6AE 0 0 0 0 0 0 0 0  6AE    NU=%T(ZE), *GO TO MP.USEDI-1;                                 2971.000
                                                                                                                    2972.000
                                                **                                                                  2973.000
                                                **      ENTRY PARAMETER :                                           2974.000
                                                **              N=@FF                                               2975.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

S E L 3 2 / 7 5 C P U T D

PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                **                                                  2976.000
                                                                                                    2977.000

                                                CLEAR.TIMEOUT                                       2978.000

0515 1 4 0 0 1 0 0 0 00 0 7A7 0 0 0 0 0 0 0 0  S17        IF IORESRDY *HOP $+2;                    2979.000

0516 A 4 3 0 1 0 0 0 00 0 905 0 0 0 0 0 0 0 0  S15        DFCRN, IF %NCTR7 *HOP $-1;               2980.000

0517 0 1 0 0 0 0 C 0 15 F 000 0 0 0 0 0 0 0 0              NU=T(7E), CLRTO, *JUMPJ;                 2981.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                2982.000
                                        ***                                                                     2984.000
                                        ***  INTERRUPT I/O ERROR 1                                              2985.000
                                        ***                                                                     2986.000
                                        ***    THIS ERROR ROUTINE IS ENTERED FROM THE INTERRUPT HANDLER          2987.000
                                        ***    FOR ADVANCE TRANSFER ERRORS OF THE FOLLOWING TYPE :              2988.000
                                        ***    1.  I/O RETRY RETRY COUNT EXHAUSTED.                             2989.000
                                        ***    2.  I/O CHANNEL BUSY RETRY COUNT EXHAUSTED.                      2990.000
                                        ***    3.  I/O NO RESPONSE.                                             2991.000
                                        ***    4.  I/O TIMEOUT. (THIS IS A ILLOGICAL ERROR FOR ADVANCE XFER)    2992.000
                                        ***                                                                     2993.000

                                                                                                                2994.000

                                        (@520)                                                                  2995.000

                                        INT.IO.ERR1                                                             2996.000

0520 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0            T=@00000000; CLEAR T REG                               2997.000

                                        INT.IO.ERR3                                                             2998.000

0521 9 4 0 0 1 0 0 0 00 0 783 0 0 0 0 0 0 0 0  523       IF %IONORESP *HOP CK.TOTIMEOUT;                        2999.000

0522 0 4 0 0 1 E 1 0 02 0 508 0 0 0 0 0 0 0 0  528       S=(INT.ERR.FLG:N.RESP.FLG:INT.ERR.FLG), *HOP INT.ERR.EXIT1;  3000.000

                                        CK.IOTIMEOUT                                                            3001.000

0523 9 4 0 0 1 0 0 0 00 0 085 0 0 0 0 0 0 0 0  525       IF %IOTIMEOUT *HOP CK.BUSY1;                           3002.000

0524 0 4 0 0 1 E 1 0 02 0 428 0 0 0 0 0 0 0 0  528       S=(INT.ERR.FLG:IO.TIM.FLG:INT.ERR.FLG), *HOP INT.ERR.EXIT1;  3003.000

                                        CK.BUSY1                                                                3004.000

0525 9 4 0 0 1 0 0 0 00 0 887 0 0 0 0 0 0 0 0  527       IF %IOCHBUSY *HOP CK.RETRY1;                           3005.000

0526 0 4 0 0 1 E 1 0 02 0 498 0 0 0 0 0 0 0 0  528       S=(INT.ERR.FLG:CNT.EXH.FLG:BUSY.FLG), *HOP INT.ERR.EXIT1;  3006.000

                                        CK.RETRY1                                                               3007.000

0527 0 0 0 0 1 E 1 0 02 0 410 0 0 0 0 0 0 0 0            S=(INT.ERR.FLG:CNT.EXH.FLG);                           3008.000

                                        INT.ERR.EXIT1                                                           3009.000

0528 0 5 0 0 0 A 1 0 00 0 0C1 0 0 0 0 0 0 0 0  5C1       S=S:T, *GO TO INT.ERR.EXIT2;                           3010.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING
PC T S M A B + B R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    3011.000
                                                                                                                    3013.000
                                                    ***                                                             3014.000
                                                    *** INTERRUPT I/O ERROR 2                                       3015.000
                                                    ***                                                             3016.000
                                                    ***    THIS ROUTINE IS ENTERED FROM THE INTERRUPT HANDLER FOR    3017.000
                                                    ***    FINAL TRANSFER ERRORS OF THE FOLLOWING TYPE :            3018.000
                                                    ***    1.  I/O CHANNEL BUSY.                                    3019.000
                                                    ***    2.  I/O NO RESPONSE.                                     3020.000
                                                    ***    3.  I/O TIMEOUT.                                         3021.000
                                                    ***    4.  I/O RETRY RETRY COUNT EXHAUSTED.                     3022.000
                                                    ***                                                             3023.000

                                                    INT.IO.ERR2                                                     3024.000
0529 0 0 0 0 1 E F 0 02 0 600 0 0 0 0 0 0 0 0            T=(INT.ERR.FLG:FINAL.FLG);                                 3025.000
052A 9 4 0 0 1 0 0 0 00 0 881 0 0 0 0 0 0 0 0  521       IF %IOCHBUSY *HOP INT.IO.ERR3;                             3026.000
052B 0 4 0 0 1 E 1 0 02 0 088 0 0 0 0 0 0 0 0  528       S=BUSY.FLG, *HOP INT.ERR.EXIT1;                            3027.000
                                                                                                                    3028.000
                                                    INT.RDY.TIMEOUT                                                 3029.000
052C 0 4 0 0 1 E 1 0 02 0 44E 0 0 0 0 0 0 0 0  52E       S=(INT.ERR.FLG:RDY.TIM.FLG), *HOP INT.DRT.TIMEOUT+1;       3030.000
                                                                                                                    3031.000
                                                    INT.DRT.TIMEOUT                                                 3032.000
052D 0 0 0 0 1 E 1 0 02 0 620 0 0 0 0 0 0 0 0            S=(INT.ERR.FLG:IO.TIM.FLG:FINAL.FLG);                      3033.000
052E 0 4 0 3 1 E F 0 02 0 008 0 0 0 0 0 0 0 0  528       T=@00000000, *HOP INT.ERR.EXIT1;                           3034.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L  3 2 / 7 5  C P U                                                    T R A P  A N D  E R R O R  S C H E D U L I N G
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                3035.000
                                                        ***                                                                     3037.000
                                                        ***  CURRENT INSTRUCTION ERROR (MEMORY TYPE)                            3038.000
                                                        ***                                                                     3039.000
                                                        ***    NOTE:  IN MOST USES OF THIS SUB-ROUTINE A SECOND MEMORY          3040.000
                                                        ***           OPERATION IS ALREADY IN PROGRESS AND WILL CLEAR THE       3041.000
                                                        ***           ERROR FLAGS WITHIN TWO CLOCKS!                            3042.000
                                                        ***                                                                     3043.000
                                                        ***    ENTRY PARAMETERS :                                               3044.000
                                                        ***    NCTR = 0 ; CPU MODE IS NORMAL AND DOES NOT NEED RESTORED.        3045.000
                                                        ***    NCTR NOT =0 ; CPU MODE HAS BEEN ALTERED AND NEEDS TO BE RESTORED 3046.000
                                                        ***              BEFORE A TRAP CONTEXT SWITCH WILL BE ACCURATE.         3047.000
                                                        ***                                                                     3048.000
    00000000000000FF                                    MEM.ERR.MSK     $EQ     @000000FF MEMORY ERROR JUMP MASK                3049.000

                                                                                                                                3050.000

                                                        (@530)                                                                  3051.000

                                                        CURRENT.INST.ERROR                                                      3052.000

0530 9 4 0 0 1 0 0 0 00 0 7D2 0 0 0 0 0 0 0 0  532          IF %OPNORESP *HOP CK.TIMEOUT;                                       3053.000

0531 0 4 0 3 1 E 1 0 02 0 488 0 0 0 0 0 0 0 0  538          S=(OPRND.NONPRESENT&MEM.ERR.MSK), *HOP CURRENT.I.EXIT;              3054.000

                                                        CK.TIMEOUT                                                              3055.000

0532 9 4 0 0 1 0 0 0 00 0 DD4 0 0 0 0 0 0 0 0  534          IF %OPTIMEOUT *HOP CK.OPRNDPE;                                      3056.000

0533 0 4 0 3 1 E 1 0 02 0 4A8 0 0 0 0 0 0 0 0  538          S=(OPERAND.TIMEOUT&MEM.ERR.MSK), *HOP CURRENT.I.EXIT;               3057.000

                                                        CK.OPRNDPE                                                              3058.000

0534 9 4 0 0 1 0 0 0 00 0 F56 0 0 0 0 0 0 0 0  536          IF %OPRNDPE *HOP UNEXPLAINED;                                       3059.000

0535 0 4 0 3 1 E 1 0 02 0 408 0 0 0 0 0 0 0 0  538          S=(OPRND.PE&MEM.ERR.MSK), *HOP CURRENT.I.EXIT;                      3060.000

                                                        UNEXPLAINED                                                             3061.000

0536 1 4 0 0 1 0 0 0 00 0 FD5 0 0 0 0 0 0 0 0  535          IF OPMINUS *HOP $-1;                                                3062.000

0537 0 0 0 3 1 E 1 0 02 0 3D0 0 0 0 0 0 0 0 0               S=(UNEXPLAINED.ERR&MEM.ERR.MSK); GEN EXPLAINED DISPLACEMENT         3063.000

                                                        CURRENT.I.EXIT                                                          3064.000

0538 0 0 0 2 1 E F 0 02 0 054 0 0 0 0 0 0 0 0               T=(TRAP.TBL&@0F00), RESET(HIREG); SET BASE ADDR OF ERR JUMP TBL     3065.000

0539 0 0 0 0 0 A 1 0 00 0 005 0 0 0 0 0 0 0 0               S=S:T, RESET(ENAUORD); COMBINE BASE & DISPLACEMENT OF ERR TBL       3066.000

053A A 5 7 0 1 0 0 0 00 0 9C4 0 0 0 0 0 0 0 0  5C4          IF %NCTRZ *LINK RESTORE.CPU.MODE; GO RESTORE MODE IF REQUIRED       3067.000

053B 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0  59C          PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC                        3068.000

053C 6 0 1 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0               *JUMPS;                                                             3069.000
```

```
PC  TSMAB+DR YX PCH+BDM+BAD ADDR

                                                                                3070.000
                                    ***                                         3071.000
                                    *** UNEXPLAINED MEMORY ERROR                3072.000
                                    ***                                         3073.000
                                    ***  ENTRY TO THIS ERROR HANDLER IS CAUSED BY A MEMORY ERROR  3074.000
                                    ***  THAT WAS ORGINALLY SENSED BY A GLOBAL TEST BUT CAN NOT   3075.000
                                    ***  BE DEFINED BY THE SPECIFIC SUB-TESTS.  THE ERROR MAY HAVE 3076.000
                                    ***  BEEN CLEARED BY A SUBSEQUENT READ OR WRITE MEMORY.       3077.000
                                    ***                                         3078.000
                                                                                3079.000
                                    UNEXPLAINED.ERR                             3080.000
053D 0 0 0 2 1 E 1 0 02 0 400 0 0 0 0 0 0 0 0       S=UNEXP.M.ERR.FLG;          3081.000
053E 0 5 0 0 1 0 0 0 00 0 078 0 0 0 0 0 0 0 0 578   *GO TO MACHINE.CHECK.STATUS; 3082.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                3083.000
                                                    ***                                                         3085.000
                                                    *** TRAP TABLE                                              3086.000
                                                    ***                                                         3087.000
                                                    ***  THIS TRAP/ERROR TABLE MAY BE ENTERED BY DIRECT BRANCHES FROM   3088.000
                                                    ***  EXTERNAL EVENT PROCESSING OR DIRECT PARSING OF ERROR CONDITIONS 3089.000
                                                    ***  VIA THE JUMPS FUNCTION.                                3090.000
                                                    ***                                                         3091.000

                                                                                                                3092.000
                                                    TRAP.TBL(@540)                                              3093.000

                                                    OPRND.PE                                                    3094.000

0540 0 4 0 1 1 E 1 0 02 0 803 0 0 0 0 0 0 0 0  543      S=OP.PE.FLG, *HOP MEM.PE; CPU STATUS OPERAND FETCH PE      3095.000

                                                    INST.PE                                                     3096.000

0541 6 0 0 2 4 A B 7 02 0 010 0 0 0 0 0 0 0 0           FR(TRACE)=@00000100:FR(TRACE);  INDICATE INSTR SKIP        3097.000

0542 0 0 0 1 1 E 1 0 02 0 400 0 0 0 0 0 0 0 0           S=INST.PE.FLG; SET CPU STATUS INSTRUCTION FETCH PE          3098.000

                                                    MEM.PE                                                      3099.000

0543 0 0 0 3 1 E E 0 02 0 000 0 0 0 0 0 0 0 0           FULLMAR=@00000000; CLEAR MAR                               3100.000

0544 0 0 0 0 1 0 0 0 1C 0 C84 0 0 0 0 0 0 0 0           READ, SET(DPEFF); CLEAR ERROR & SET PARITY ERROR LITE      3101.000

0545 0 0 0 1 1 E F 0 02 0 120 0 0 0 0 0 0 0 0           T=@00120000; SET PSEUDO INTERRUPT LEVEL FOR MPE            3102.000

0546 0 5 0 0 1 0 0 0 15 0 063 0 0 0 0 0 0 0 0  563      CLRTO, *GO TO QUEUE.TEST;                                  3103.000

                                                    OPERAND.NONPRESENT                                          3104.000

0547 1 4 0 0 1 0 0 0 00 0 0DA 0 0 0 0 0 0 0 0  54A      IF OPTIMEOUT *HOP OPERAND.TIMEOUT;                         3105.000

                                                    OPRND.NONPRESENT                                            3106.000

0548 0 0 0 3 1 E 0 0 02 0 001 0 0 0 0 0 0 0 0           NOD=@00000000, SETCC(AL); FAKE CC SET FOR ZERO DATA        3107.000

0549 0 4 0 1 1 E 1 0 02 0 20D 0 0 0 0 0 0 0 0  54D      S=OP.NPM.FLG, *HOP NON.PRESENT.MEMORY; SET CPU STATUS ERR   3108.000

                                                    OPERAND.TIMEOUT                                             3109.000

054A 0 4 0 1 1 E 1 0 02 0 24D 0 0 0 0 0 0 0 0  54D      S=(TIM.FLG:OP.NPM.FLG), *HOP NON.PRESENT.MEMORY; SET CPU   3110.000
                                                    *       STATUS TIMEOUT AND NON-PRESENT MEMORY FLAGS.        3112.000

                                                    INSTRUCTION.NONPRESENT                                      3114.000

054B 0 4 0 1 1 E 1 0 02 0 10D 0 0 0 0 0 0 0 0  54D      S=INST.NPM.FLG, *HOP NON.PRESENT.MEMORY; SET CPU STATUS ERR 3115.000

                                                    INSTRUCTION.TIMEOUT                                         3116.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
054C 0 0 0 1 1 E 1 0 02 0 140 0 0 0 0 0 0 0 0          S=(TIM.FLG:INST.NPM.FLG) ;SET CPU STATUS TIMEOUT AND NPM FLAGS   3117.000
                                                  NON.PRESENT.MEMORY                                                    3118.000
054D 0 0 0 3 1 E E 0 02 0 000 0 0 0 0 0 0 0 0          FULLMAR=@00000000;                                               3119.000
054E 0 0 0 0 1 0 0 0 1C 0 080 0 0 0 0 0 0 0 0          READ; CLEAR ERRORS                                               3120.000
054F 0 0 0 1 1 E F 0 02 0 240 0 0 0 0 0 0 0 0          T=@00240000; SET PSEUDO INTERRUPT LEVEL FOR NON-PRESENT          3121.000
0550 9 4 0 0 1 0 0 0 15 0 F66 0 0 0 0 0 0 0 0  556     CLRTO, IF %MAPINVALID *GO TO QUEUE.TEST.BRANCH;                  3122.000
                                                  PRIVILEGE.VIOLATION                                                   3123.000
0551 6 0 0 0 4 B B 7 02 0 087 0 0 0 0 0 0 0 0          FR(TRACE)=%@0F000000&FR(TRACE),CLRS; RESET PRIV VIOL FLAG        3124.000
0552 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0          SET(ENAUORD);                                                    3125.000
0553 0 0 0 1 1 E F 0 02 4 260 0 0 0 0 0 0 0 0          T=@00260000,RSTPROTV; SET PSEUDO INTERRUPT LEVEL FOR PRIV.VIOL   3126.000
0554 1 4 0 0 1 0 0 0 00 0 F67 0 0 0 0 0 0 0 0  557     IF MAPINVALID *HOP MAP.FAULT.TRAP;                               3127.000
0555 0 6 0 0 1 0 0 0 00 0 2D0 0 0 0 0 0 0 0 0  2D0     *GOTO PRIV.TYPE;                                                 3128.000
                                                  QUEUE.TEST.BRANCH                                                     3129.000
0556 0 5 0 0 1 0 0 0 00 0 063 0 0 0 0 0 0 0 0  563     *GO TO QUEUE.TEST;                                               3130.000
                                                  MAP.FAULT.TRAP                                                        3131.000
0557 0 0 0 2 1 E 1 0 02 0 085 0 0 0 0 0 0 0 0          S=MAP.INVALID.FLG, RESET(ENAUORD); SET CPU STATUS ERR FLG        3132.000
0558 0 5 7 0 1 E F 0 00 0 096 0 0 0 0 0 0 0 0  596     T=S, *LINK UPDATE.CPU.STATUS;                                    3133.000
0559 0 0 0 3 1 E F 0 02 0 8C0 0 0 0 0 0 0 0 0          T=@0000008C;                                                     3134.000
055A 0 4 0 2 0 A 1 0 02 0 01F 0 0 0 0 0 0 0 0  55F     S=@00000100:T, *HOP BLK.ERR.EXIT; GEN 18C INTR. VECTOR LOC       3135.000
                                                  BLK.MODE.ERR                                                          3136.000
055B 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0  59C     PCTOMAR, *LINK DECREMENT.PC; ENTER THIS ERROR FROM A WAIT INST.  3137.000
                                                  BLOCK.MODE.TIMEOUT                                                    3138.000
055C 0 0 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0          SET(UNBLOCK); CLEAR BLOCK MODE ERROR                             3139.000
055D 0 0 0 0 1 0 0 0 00 0 106 0 0 0 0 0 0 0 0          RESET(UNBLOCK);  RETURN TO BLOCK MODE                            3140.000
055E 0 0 0 3 1 E 1 0 02 0 E40 0 0 0 0 0 0 0 0          S=@000000E4; SET INTERRUPT VECTOR LOCATION FOR BLOCK ERROR TRAP  3141.000
                                                  BLK.ERR.EXIT                                                          3142.000
055F 0 6 0 0 1 0 0 0 00 0 2CD 0 0 0 0 0 0 0 0  2CD     *GOTO MAP.FAULT.TYPE;       DETERMINE IPU/CPU MODE               3143.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                3144.000
                                                UNDEFINED.INSTRUCTION                                           3145.000
0560 6 0 0 0 4 B B 7 02 0 047 0 0 0 0 0 0 0 0       FR(TRACE)=%@04000000&FR(TRACE),CLRS; RESET UNDEF INSTR FLAG 3146.000
0561 0 4 0 1 1 E F 0 02 0 253 0 0 0 0 0 0 0 0 563  T=@00250000, *HOP QUEUE.TEST; SET UNDEF PSEUDO INT. LEVEL  3147.000
                                                **                                                              3148.000
                                                **                                                              3149.000
                                                (@563)                                                          3150.000
                                                QUEUE.TEST                                                      3151.000
0563 0 0 0 0 0 0 8 0 00 0 A00 0 0 0 0 0 0 0 0       R(INTR)=T; SAVE PSEUDO INTERRUPT LEVEL                      3152.000
0564 0 5 7 0 1 E F 0 00 D 096 0 0 0 0 0 0 0 0 596  T=S, OTHERBANK, *LINK UPDATE.CPU.STATUS;                    3153.000
0565 0 6 7 0 1 0 0 0 00 0 28A 0 0 0 0 0 0 0 0 28A  *LINK INSTR.SKIP; GO ADJUST PC TO POINT AT THE ERROR +1W     3154.000
0566 6 0 0 1 4 A 1 2 02 0 800 0 0 0 0 0 0 0 0      S=@00800000:FR(INTR), OTHERBANK;                            3155.000
0567 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0      S=SCRATCH(S);    FETCH INTERRUPT ENTRY                      3156.000
0568 0 6 0 0 1 0 0 0 00 0 F6A 0 0 0 0 0 0 0 0 F6A    *GOTO TEST.IPU.CPU;                TEST MODE              3157.000
                                                TRAP.MODE55                                                     3158.000
0569 0 0 0 0 0 D 0 0 02 0 010 0 0 0 0 0 0 0 0      NOD=@01000000&T;         TEST IF ENTRY ENABLED              3159.000
056A 0 0 0 0 0 0 D 0 0 02 0 040 0 0 0 0 0 0 0 0    NOD=@04000000&T;         TEST IF ENTRY ALREADY ACTIVE       3160.000
056B B 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0      IF ALUZ *JUMPJ;                                             3161.000
056C C 1 0 0 4 0 1 0 00 D A00 0 0 0 0 0 0 0 0      S=R(INTR),OTHERBANK,IF NALUZ *JUMPJ;                        3162.000
056D 0 0 0 0 1 0 0 0 00 0 F04 0 0 0 0 0 0 0 0      SET(ENAINTFF);         ENABLE INTERRUPTS                    3163.000
056E 0 0 0 1 6 D F 0 02 0 7F0 0 0 0 0 0 0 0 0      T=@007F0000&INTLVL;  GET POLLING INTERRUPT LEVEL            3164.000
056F 9 4 0 0 1 0 0 0 00 0 DA1 0 0 0 0 0 0 0 0 571  IF %FFINT *60 TO $+2;    INTERRUPT PENDING USE ALEVEL     3165.000
0570 6 0 0 1 4 D F 4 02 0 7F0 0 0 0 0 0 0 0 0      T=@007F0000&FR(ALEVEL); GET HIGHEST ACTIVE LEVEL            3166.000
0571 0 0 0 0 0 5 C 0 00 0 704 0 0 0 0 0 0 0 0      NU=S-T,RESET(ENAINTFF);         TEST IT ACT INTR LT REQ     3167.000
                                                CALM.TRAP                                                       3168.000
                                                PF.QUEUE                                                        3169.000
0572 0 0 0 0 4 0 F 0 00 0 900 0 0 0 0 0 0 0 0      T=R(STMASK);                                                3170.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0573 0 0 0 5 0 D 6 0 00 0 804 0 0 0 0 0 0 0 0          DI=STATUSRT, SET(HIREG); GET PSW STATUS                          3171.000

0574 E 1 0 0 4 0 1 2 00 D D00 0 0 0 0 0 0 0 0          S=FR(INTR), OTHERBANK, IF %NCTR0 *JUMPJ;                         3172.000

0575 6 0 0 0 4 A B 7 02 D 100 0 0 0 0 0 0 0 0          FR(TRACE)=@10000000:FR(TRACE),OTHERBANK;  SET TRAP FLAG          3173.000

0576 0 6 0 0 3 0 F 0 00 D 047 0 0 0 0 0 0 0 0   47     T=DI, OTHERBANK, *GO TO INTERRUPT;                               3174.000

                                                       MACHINE.CHECK.TRAP                                               3175.000

0577 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0   59C    PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC                     3176.000

                                                       MACHINE.CHECK.STATUS                                             3177.000

0578 0 5 7 0 1 E F 0 00 0 096 0 0 0 0 0 0 0 0   596    T=S, *LINK UPDATE.CPU.STATUS;  GO POST ERROR STATUS              3178.000

0579 0 0 0 2 1 E F 0 02 0 014 0 0 0 0 0 0 0 0          T=@00000100, RESET(HIREG);                                       3179.000

057A 0 0 0 3 0 A 1 0 02 0 840 0 0 0 0 0 0 0 0          S=@000000B4:T; GENERATE 184 INTERRUPT VECTOR LOCATION            3180.000

057B 0 5 0 0 1 0 0 0 00 0 080 0 0 0 0 0 0 0 0   580    *GO TO TRAP.MODE75;                                              3181.000

                                                                                                                        3182.000
                                                       **                                                               3183.000
                                                       **                                                               3184.000
                                                       ** EXIT TO SYSTEM CHECK TRAP (CURRENT INSTRUCTION ERROR)        3185.000
                                                       **                                                               3186.000
                                                       **     ENTRY PARAMETERS FOR SYSTEM.CHECK.STATUS:                 3187.000
                                                       **        S = CPU STATUS FLAGS TO BE POSTED                      3188.000
                                                       **                                                               3189.000
                                                       **     ENTRY PARAMETERS FOR SYSTEM.CHECK.TRAP:                   3190.000
                                                       **        PC = EXECUTION ADDRESS PLUS 2 WORDS                    3191.000
                                                       **                                                               3192.000
                                                       **        IF RHFLAG=1, CURRENT INSTRUCTION WAS A LEFT HALFWORD INST.  3193.000
                                                       **                                                               3194.000
                                                       **        IF FLAG=1, CURRENT INSTRUCTION WAS A LEFT HALFWORD INST  3195.000
                                                       **                                                               3196.000
                                                       **        IF RHFLG=0, AND FLAG=0, CURRENT INSTRUCTION WAS FULLWORD OR  3197.000
                                                       **          A RIGHT HALFWORD INSTRUCTION                         3198.000
                                                       **                                                               3199.000

                                                                                                                        3200.000

                                                       SYSTEM.CHECK.TRAP                                                3201.000

057C 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0   59C    PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC                     3202.000

                                                       SYSTEM.CHECK.STATUS                                              3203.000

057D 0 5 7 0 1 E F 0 00 0 096 0 0 0 0 0 0 0 0   596    T=S, *LINK UPDATE.CPU.STATUS;                                    3204.000

057E 0 0 0 2 1 E F 0 02 0 014 0 0 0 0 0 0 0 0          T=@00000100, RESET(HIREG);                                       3205.000

057F 0 0 0 3 0 A 1 0 02 0 880 0 0 0 0 0 0 0 0          S=@00000088:T; SET SYSTEM CHECK TRAP DEDICATED ADDRESS           3206.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SFL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                                3207.000
                                                                TRAP.MODE75                                                     3208.000
0580 6 0 0 2 4 A B 7 02 0 040 0 0 0 0 0 0 0 0                       FR(TRACE)=@00000400:FR(TRACE);  SET 75 TRAP FLAG            3209.000
0581 0 0 0 0 1 E F 0 00 0 305 0 0 0 0 0 0 0 0                       T=S, RESET(MAPMODE);   EXIT MAP MODE                        3210.000
0582 0 0 0 0 0 0 E 0 00 F 00D 0 0 0 0 0 0 0 0                       FULLMAR=T(ZE), CLDNU;  LOAD SI VECTOR ADDR                  3211.000
0583 0 0 0 0 1 0 0 0 1C 1 884 0 0 0 0 0 0 0 0                       READ, FRCWORD, SET(HIREG); FETCH SI VECTOR                  3212.000
0584 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0                       S=SCRATCH(@91);  FETCH CPU STATUS WORD                      3213.000
0585 6 0 0 0 4 A B 7 02 D 100 0 0 0 0 0 0 0 0                       FR(TRACE)=@10000000:FR(TRACE), OTHERBANK; SET TRAP FLAG     3214.000
0586 0 0 0 0 1 E 8 0 00 D 800 0 0 0 0 0 0 0 0                       R(TBL2)=S, OTHERBANK;  SAVE CPU STATUS WORD                 3215.000
0587 0 0 0 3 0 0 F 0 02 0 FF0 0 0 0 0 0 0 0 0                       T=@000000FF&T; CLEAN UP ERROR STATUS                        3216.000
0588 0 0 0 1 0 0 4 0 02 A 910 0 0 0 0 0 0 0 0                       SCRATCH(@91)=T;  SAVE UPDATED STATUS WORD                   3217.000
0589 4 4 0 3 1 E 1 0 02 8 40R 0 0 0 0 0 0 0 0  58B                  S=@00000040, IF UNBLOCK *GO TO $+2; SET DISABLE TRAP MASK   3218.000
058A 0 6 7 0 1 0 0 0 00 0 FEB 0 0 0 0 0 0 0 0  FEB                  *LINK SET.BLOCKED.M.FLAG;                                   3219.000
058B 0 0 0 0 4 D 0 0 00 0 800 0 0 0 0 0 0 0 0                       NOD=S&R(TBL2);  TEST FOR DISABLE TRAP FLAG IN CPU STATUS     3220.000
058C 0 0 0 0 4 0 F 0 00 D 900 0 0 0 0 0 0 0 0                       T=R(STMASK), OTHERBANK;  FETCH STATUS MASK                  3221.000
058D C 5 0 0 1 0 0 0 00 0 0A3 0 0 0 0 0 0 0 0  5A3                  IF NALUZ *GO TO M.75.TRAP.HALT;                             3222.000
                                                                SVC.TRAP.ENTRY.POINT                                            3223.000
058E 0 0 0 5 0 D 8 0 00 0 204 0 0 0 0 0 0 0 0                       R(CPSTS)=STATUS&T, RESET(PRIV); GET CURRENT CC'S & EXT BIT   3224.000
058F 0 0 0 3 3 3 E 0 02 0 0C0 0 0 0 0 0 0 0 0                       FULLMAR=@0000000C+DI;  VECTOR TO N.PSW2                     3225.000
0590 0 0 0 0 3 0 8 0 1C 1 680 0 0 0 0 0 0 0 0                       R(SI.VECTOR)=DI, READ, FRCWORD; SAVE SI VECTOR & FETCH N.PSW2 3226.000
0591 9 4 0 0 1 0 0 0 00 0 553 0 0 0 0 0 0 0 0  593                  IF %OPNORESP:OPRNDPF:OPTIMEOUT *HOP $+2;                    3227.000
                                                                INTR.ERR9                                                       3228.000
0592 0 5 0 0 0 0 C 0 00 F 030 0 0 0 0 0 0 0 0  530                  NU=T(ZE), *GO TO CURRENT.INST.ERROR;                        3229.000
0593 6 0 0 4 1 E B 1 02 0 000 0 0 0 0 0 0 0 0                       FR(RDEV)=@00FFFFFF;   CREATE DUMMY DVC ENTRY                3230.000
0594 0 0 0 0 3 0 8 0 00 0 000 0 0 0 0 0 0 0 0                       R(N.PSW2)=DI; SAVE NEW PSW2                                 3231.000
0595 0 6 0 0 1 0 0 0 00 8 06A 0 0 0 0 0 0 0 0   6A                  PCTOMAR, *GO TO SCHEDULE.75.TRAP;  GET CURRENT PC IN MAR    3232.000
                                                                                                                                3233.000
                                                                **                                                              3234.000
```

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                ** UPDATE CPU STATUS WORD IN SCRATCH PAD ADDR 91                          3235.000
                                                **     ENTRY PARAMETERS:                                                 3236.000
                                                **       T = NEW STATUS FLAGS                                            3237.000
                                                **                                                                       3238.000

                                                                                                                         3239.000

                                                UPDATE.CPU.STATUS                                                        3240.000

0596 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0           S=SCRATCH(@91); FETCH CPU STATUS                                 3241.000

0597 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0           T=S:T; MERGE CURRENT AND NEW STATUS                              3242.000

0598 0 0 0 7 0 D 1 0 00 0 500 0 0 0 0 0 0 0 0           S=@FFFFFF50&T; SAVE MERGED ERROR STATUS & DISABLE TRAP BIT       3243.000

0599 0 6 7 0 1 0 0 0 00 0 FB7 0 0 0 0 0 0 0 0  FB7      *LINK COMPUTE.CPU.STATUS;  GO COMPUTE MODE STATUS                 3244.000

059A 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0           T=S:T; MERGE ERROR STATUS AND MODE STATUS                        3245.000

059B 0 1 0 1 0 0 4 0 02 A 910 0 0 0 0 0 0 0 0           SCRATCH(@91)=T, *JUMPJ; POST CPU STATUS & RETURN                 3246.000

                                                                                                                         3247.000
                                                **                                                                       3248.000
                                                ** DECREMENT PC FOR CURRENT INSTRUCTION ERROR                            3249.000
                                                **                                                                       3250.000
                                                **     ENTRY PARAMETERS:                                                 3251.000
                                                **        PC = MAR = CURRENT EXECUTION ADDRESS PLUS 2 WORDS              3252.000
                                                **        IF FLAG AND RHFLG=0, CURRENT INSTRUCTION WAS A FULLWORD OR     3253.000
                                                **          RIGHT HALFWORD INSTRUCTION                                   3254.000
                                                **        IF FLAG OR RHFLG=1, CURRENT INSTRUCTION WAS A LEFT HALFWORD    3255.000
                                                **          INSTRUCTION, FOLLOWED BY A NON 'NOP' RIGHT HALFWORD          3256.000
                                                **          INSTRUCTION THAT HAS NOT BEEN EXECUTED.                      3257.000
                                                **                                                                       3258.000
                                                **     EXIT PARAMETERS:                                                  3259.000
                                                **        IF FLAG AND RHFLG=0, PC = CURRENT PC -4 (-1W)                  3260.000
                                                **                                                                       3261.000
                                                **        IF FLAG OR RHFLG=1, PC = CURRENT PC -8 (-2W) AND R(TRACE)      3262.000
                                                **          RIGHT HALFWORD INDICATOR IS SET (BIT 16)                     3263.000

                                                                                                                         3264.000

                                                DECREMENT.PC                                                             3265.000

059C 0 0 0 7 1 3 2 0 00 0 FC0 0 0 0 0 0 0 0 0           PC=@FFFFFFFC+MAR; BACKDATE PC BY 4 (1W)                          3266.000

059D A 4 0 0 1 0 0 0 00 0 A0F 0 0 0 0 0 0 0 0  59F      IF %RHFLAG *GO TO $+2;                                           3267.000

059E 0 5 0 0 1 0 0 0 00 0 0A0 0 0 0 0 0 0 0 0  5A0      *GO TO RIGHT.HALFWORD;                                           3268.000

059F 9 1 0 0 1 0 0 0 00 0 0E0 0 0 0 0 0 0 0 0           IF %FLAG*JUMPJ; EXIT IF NO RIGHT HAND FLAGS                      3269.000

                                                RIGHT.HALFWORD                                                           3270.000

05A0 0 0 0 7 1 3 2 0 00 0 F80 0 0 0 0 0 0 0 0           PC=@FFFFFFF8+MAR; BACKDATE PC BY 8 (2W)                          3271.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING
PC TSMAB+DK YX PCH+BDM+BAD ADDR

```
                                        SET.RHFLAG.IND                                                  3272.000

05A1 6 0 0 2 4 A B 7 02 0 800 0 0 0 0 0 0 0 0       FR(TRACE)=@00008000:FR(TRACE); SET HALFWORD INDICATOR       3273.000

05A2 0 1 0 0 1 0 0 0 0C 0 705 0 0 0 0 0 0 0 0       RSTRHF, RESET(FLAG), *JUMPJ;                                3274.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                                                    3275.000
                                        ***                                                                         3277.000
                                        *** MODE 75 TRAP HALT                                                       3278.000
                                        ***                                                                         3279.000
                                        ***  NOTE: THE FOLLOWING ERROR SEQUENCE IS ENTERED WHEN A 75 MODE           3280.000
                                        ***        TRAP IS ENCOUNTERED AND THE 75 MODE TRAPS HAVE NOT BEEN          3281.000
                                        ***        ENABLED BY SOFTWARE.  SINCE 75 MODE TRAPS ARE NON-DEFERABLE,     3282.000
                                        ***        THE CPU IS HALTED WITH THE FOLLOWING DISPLAYS :                  3283.000
                                        ***    1.  PSW 1 DISPLAYS THE TRAP/INTERRUPT(IVL) FOR THE TRAP CAUSING     3284.000
                                        ***        THE HALT.                                                        3285.000
                                        ***    2.  THE INTERRUPTED PSW 1 IS STORED AT MEMORY LOCATION 530.          3286.000
                                        ***    3.  THE INTERRUPTED PSW 2 IS STORED AT MEMORY LOCATION 534.          3287.000
                                        ***    4.  THE CURRENT CPU STATUS WORD IS STORED AT MEMORY LOCATION 538     3288.000
                                        ***    5.  THE CONTENTS OF R(RDEV) IS STORED AT 53C.                        3289.000
                                        ***    6.  THE INITIAL CONTENTS OF R(INTRTAB) IS STORED AT 540.             3290.000
                                        ***    7.  THE INTERRUPT ACTIVE LITE IS TURNED ON TO INDICATE A             3291.000
                                        ***        ERROR CONDITION.                                                 3292.000
                                        ***                                                                         3293.000
                                        ***  NOTE:  THESE MEMORY LOCATIONS ARE RESERVED FOR CPU POWER FAIL          3294.000
                                        ***         SCRATCH PAD ROLLOUT, AND SHOULD NOT OVERLAY A SYSTEM-           3295.000
                                        ***         BOOT SOFTWARE PROGRAM.                                          3296.000
                                        ***                                                                         3297.000

                                                                                                                    3298.000
                                        M.75.TRAP.HALT                                                              3299.000
05A3 0 6 0 0 1 0 0 0 00 0 C73 0 0 0 0 0 0 0 0  C73          *GOTO IPU.TRAP.CHECK;                                   3300.000
                                        *                                                                           3301.000
                                        M.75.TRAP.RTN                                                               3302.000
                                        *                                                                           3303.000
05A4 0 0 0 0 1 0 2 0 00 8 000 0 0 0 0 0 0 0 0          PC=MAR, PCTOMAR; SAVE TRAP LVL AND GET CURRENT PC            3304.000
05A5 6 0 0 1 4 A 1 0 02 D FF0 0 0 0 0 0 0 0 0          S=@00FF0000:FR(PCMASK), OTHERBANK; SET S = @00FFFFFC         3305.000
05A6 4 6 7 0 1 D F 0 00 C 1FF 0 0 0 0 0 0 0 0  1FF     T=S%MAR, IF MODE75S *LINK FORMAT.75.AEXP; GET PC FOR PSW1    3306.000
05A7 0 0 0 2 1 E 6 0 02 0 050 0 0 0 0 0 0 0 0          DI=@00000500;                                                3307.000
05A8 0 0 0 3 3 A E 0 02 0 300 0 0 0 0 0 0 0 0          FULLMAR=@00000030:DI; SET MEMORY ADDR 530                    3308.000
05A9 0 0 0 4 0 A F 0 1E 1 280 0 0 0 0 0 0 0 0          T=R(CPSTS):T, WRITE, FRCWORD; COMBINE PRIV BIT, CC'S & PC    3309.000
                                        *                 AND STORE PSW 1 AT LOCATION 530.                      3311.000
05AA 0 0 0 1 1 0 1 0 02 A 900 0 0 0 0 0 0 0 0          S=SCRATCH(@90); FETCH CURRENT PSW 2                          3313.000
05AB 0 0 0 3 1 3 E 0 02 0 044 0 0 0 0 0 0 0 0          FULLMAR=@00000004+MAR, RESET(HIREG); INCREMENT TO LOCATION 534  3314.000
05AC 0 0 0 0 1 E F 0 1F 1 080 0 0 0 0 0 0 0 0          T=S, WRITE, FRCWORD; STORE PSW 2 AT 534                      3315.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

05AD 6 4 0 0 4 0 1 0 00 D 30F 0 0 0 0 0 0 0 0  5AF        S=FR(TBL2), OTHERBANK, IF MODE75 *GO TO $+2;GET CPU STATUS WORD  3316.000
05AE 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0              S=SCRATCH(@91); GET 55 MODE STATUS WORD                  3317.000
05AF 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0              FULLMAR=@00000004+MAR;                                   3318.000
05B0 0 0 0 0 1 E F 0 1E 1 080 0 0 0 0 0 0 0 0              T=S, WRITE, FRCWORD; STORE CPU STATUS AT 538              3319.000
05B1 0 0 0 0 4 0 1 0 00 D 900 0 0 0 0 0 0 0 0              S=R(RDEV), OTHERBANK; FETCH SCRATCH PAD DEVICE ENTRY      3320.000
05B2 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0              FULLMAR=@00000004+MAR; INCREMENT TO LOCATION 53C          3321.000
05B3 0 0 0 0 1 E F 0 1E 1 080 0 0 0 0 0 0 0 0              T=S, WRITE, FRCWORD; STORE R(RDEV) AT 53C                 3322.000
05B4 0 0 0 0 4 0 1 0 00 D D04 0 0 0 0 0 0 0 0              S=R(INTRLOC), OTHERBANK, SET(DINTRA); TURN ON INTERRUPT ACTIVE  3323.000
05B5 0 0 0 0 1 0 1 0 00 A 000 0 0 0 0 0 0 0 0              S=SCRATCH(S); FETCH INITIAL INTERRUPT ENTRY               3324.000
05B6 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0              FULLMAR=@00000004+MAR; INCREMENT TO 540                   3325.000
05B7 0 0 0 0 1 E F 0 1E 1 080 0 0 0 0 0 0 0 0              T=S, WRITE, FRCWORD;          STORE R(INTRTAB) AT 540     3326.000
05B8 0 6 0 0 1 0 0 0 00 0 0CE 0 0 0 0 0 0 0 0  CE          *GO TO PANEL.HALT;                                        3327.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRAP AND ERROR SCHEDULING
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    3328.000
                                                    ***                                                             3330.000
                                                    *** MODE 55 TRAP HALT                                           3331.000
                                                    ***                                                             3332.000
                                                    ***  THIS ERROR ROUTINE IS ENTERED IF A I/O ERROR IS ENCOUNTERED 3333.000
                                                    ***  IN THE INTERRUPT HANDLER STORE, PLACE, & BRANCH ROUTINE.   3334.000
                                                    ***  A I/O ERROR IS THIS SEQUENCE IS NON-RECOVERABLE AND USUALLY 3335.000
                                                    ***  RESULTS IN A INTERRUPT ACTIVE HANG-UP.  NO ERROR REPORTING 3336.000
                                                    ***  MECHANISM IS PROVIDED IN 55 MODE FOR THIS TYPE OF ERROR, SO 3337.000
                                                    ***  THE CPU IS HALTED IN ORDER TO FLAG THE ERROR.              3338.000
                                                    ***                                                             3339.000
                                                    ***  AN ALTERNATE ENTRY POINT IS PROVIDED FOR A BRT I/O         3340.000
                                                    ***  ERROR, WHICH CAUSES THE CURRENT PC TO BE DECREMENTED ACCORDING 3341.000
                                                    ***  TO STANDARD RULES.  THE BRI I/O ERROR IS ALSO NORMALLY FATAL. 3342.000
                                                    ***                                                             3343.000
                                                                                                                    3344.000
                                                    BRI.55.HALT                                                     3345.000
                                                    M.55.TRAP.HALT                                                  3346.000
05B9 0 5 7 0 1 E F 0 00 0 096 0 0 0 0 0 0 0 0 596       T=S, *LINK UPDATE.CPU.STATUS;                               3347.000
05BA 0 0 0 3 1 E E 0 02 0 844 0 0 0 0 0 0 0 0           FULLMAR=@00000084,SET(HIREG);                               3348.000
05BB 0 0 0 2 1 A E 0 02 0 010 0 0 0 0 0 0 0 0           FULLMAR=@00000100:MAR; GENERATE MACHINE CHECK TRAP IVL      3349.000
05BC 0 5 7 0 0 0 6 0 00 0 806 0 0 0 0 0 0 0 0 506       R(TBL2)=T, *LINK DUD.50X;                                   3350.000
05BD 0 5 0 0 4 0 F 0 00 D 9A3 0 0 0 0 0 0 0 0 5A3       T=R(STMASK), OTHERBANK, *GO TO M.75.TRAP.HALT;              3351.000
                                                    BRI.ERROR                                                       3352.000
05BE 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0 59C       PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC                3353.000
05BF 0 0 0 2 1 E F 0 02 0 200 0 0 0 0 0 0 0 0           T=BRI.ERR.FLG;                                              3354.000
05C0 0 5 0 0 1 0 0 0 00 0 02A 0 0 0 0 0 0 0 0 52A       *GO TO INT.IO.ERR2+1;                                       3355.000
                                                    INT.ERR.EXIT2                                                   3356.000
05C1 0 5 7 0 0 1 C 0 00 F 015 0 0 0 0 0 0 0 0 515       NU=%T(ZE), *LINK CLEAR.TIMEOUT; GO CLEAR I/O ERRORS         3357.000
05C2 A 5 0 0 1 0 0 0 00 0 3B9 0 0 0 0 0 0 0 0 5B9       IF %MODE75 *GO TO M.55.TRAP.HALT;                           3358.000
05C3 0 5 0 0 1 0 0 0 00 0 07D 0 0 0 0 0 0 0 0 57D       *GO TO SYSTEM.CHECK.STATUS; EXIT TO SYSTEM CHECK TRAP       3359.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                T R A P   A N D   E R R O R   S C H E D U L I N G
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                3360.000
                                        ***                                                                                     3362.000
                                        *** RESTORE CPU MODE                                                                    3363.000
                                        ***                                                                                     3364.000
                                        ***    THIS SUB-ROUTINE RESTORES THE INITIAL CPU MODE DURING ERROR                      3365.000
                                        ***    EXIT SEQUENCES.                                                                  3366.000
                                        ***                                                                                     3367.000
                                        ***    ENTRY PARAMETERS :                                                               3368.000
                                        ***                                                                                     3369.000
                                        ***    NCTR NOT EQUAL TO ZERO                                                           3370.000
                                        ***                                                                                     3371.000
                                        ***      IF NCTR BIT 0 =0, FLAG IS REMEMBERED RIGHT HAND FLAG.                          3372.000
                                        ***           THIS EXIT IS FOR HALFWORD INSTRUCTIONS THAT                               3373.000
                                        ***           EXECUTE MEMORY TRANSFERS. (RWCS & RWCS)                                   3374.000
                                        ***                                                                                     3375.000
                                        ***           THIS EXIT IS FOR THE LMAP, TMAPR, LEM, SEM, CEMA                          3376.000
                                        ***           INSTRUCTIONS.                                                             3377.000
                                        ***      IF NCTR BIT 0 NOT =0, FLAG IS REMEMBERED MAPMODE.                              3378.000
                                        ***                                                                                     3379.000

                                                                                                                                3380.000
                                        RESTORE.CPU.MODE                                                                        3381.000
05C4 E 1 0 0 1 E B 5 00 0 D00 0 0 0 0 0 0 0 0           FR(TEMP3)=S, IF %NCTR0 *JUMPJ;                                          3382.000
                                        FLAG.EQUAL.MAPMODE                                                                      3383.000
05C5 0 0 0 0 1 0 0 0 00 0 305 0 0 0 0 0 0 0 0           RESET(MAPMODE);                                                         3384.000
05C6 9 4 0 0 1 0 0 0 00 0 0E8 0 0 0 0 0 0 0 0  5C8      IF %FLAG *GO TO $+2;  TEST FOR REMEMBERED MAP MODE                       3385.000
05C7 0 0 0 0 1 0 0 0 00 0 B05 0 0 0 0 0 0 0 0           SET(MAPMODE);  RESTORE MAP MODE                                         3386.000
05C8 0 0 0 0 1 E F 0 02 0 040 0 0 0 0 0 0 0 0           T=#04000000;  EXT BIT MASK                                              3387.000
05C9 0 0 0 4 0 D 1 0 00 0 200 0 0 0 0 0 0 0 0           S=R(CPSTS)&T, OTHERBANK;  GET EXT BIT FROM INITIAL CP STATUS            3388.000
05CA 0 6 7 5 1 E F 0 00 0 7E0 0 0 0 0 0 0 0 0  7E0      T=STATUS, *LINK ALTER.EXTENDED.STATUS; GET CURRENT STATUS                3389.000
05CB 0 1 0 0 4 0 1 0 00 0 D00 0 0 0 0 0 0 0 0           S=R(TEMP3), *JUMPJ;  RESTORE S REG                                      3390.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SUPERVISOR CALL
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                3391.000

                                                                                                                                3392.000
                                                    **                                                                          3393.000
                                                    ** SUPERVISOR CALL -SVC- SECONDARY DECODE (Q=32)                           3394.000
                                                    **                                                                          3395.000
                                                    **      ENTRY PARAMETERS:                                                   3396.000
                                                    **         IO 20-31 CONTAIN THE SVC CALL NUMBER.                            3397.000
                                                    **         IO 16-19 CONTAIN THE SVC VECTOR POINTER.                         3398.000
                                                    **         THE PRIMARY VECTOR ADDRESS IS STORED AT LOC @180.                3399.000
                                                    **         THE SECONDARY VECTOR IS OBTAINED FROM THE MEMORY LOC             3400.000
                                                    **         ADDRESSED BY THE PRIMARY VECTOR PLUS THE SVC VECTOR POINTER.     3401.000
                                                    **                                                                          3402.000

                                                                                                                                3403.000

                                                    (@5CC)                                                                      3404.000

                                                    SVC                                                                         3405.000

05CC 6 0 0 2 2 D B 3 02 0 F00 0 0 0 0 0 0 0 0                       FR(TEMP1)=@0000F000&IO; GET SVC TABLE POINTER                   3406.000

05CD 0 6 0 0 1 0 0 0 00 0 F7B 0 0 0 0 0 0 0 0  F7B                    *GOTO SVC.CHECK;                                             3407.000

                                                    UNDEF.55.5XX                                                                3408.000

05CE 0 6 0 0 1 0 0 0 00 0 7F7 0 0 0 0 0 0 0 0  7F7                  *GO TO UNDEF.55; SVC IS UNDEFINED IN THE 55 MODE               3409.000

05CF 0 0 0 2 1 E E 0 00 0 705 0 0 0 0 0 0 0 0                       FULLMAR=SNIBL,RESET(FLAG);SET MAR = @00000180 (SVC DEDICATED LOC) 410.000

05D0 0 0 0 0 5 0 F 0 1C 1 B87 0 0 0 0 0 0 0 0       SVC.1           T=R(TEMP1,HWS), READ, FRCWORD,CLRS; FETCH SI VECTOR AND MOVE   3411.000

                                                    *                     SVC POINTER TO T REG BITS 0-3                         3412.000

05D1 0 0 0 2 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0                       S=SNIBL;  MOVE SVC POINTER TO S REG 28-31                       3413.000

05D2 0 0 0 7 1 9 F 0 01 0 000 0 0 0 0 0 0 0 0                       SHIFTS(SLL), T=?@FFFFFF00; *2 TO SVC POINTER                    3414.000

05D3 0 0 0 2 0 A F 0 02 0 0F0 0 0 0 0 0 0 0 0                       T=@00000F00:T;  SET T = @00000FFF                               3415.000

05D4 0 5 7 0 1 0 0 0 00 8 09C 0 0 0 0 0 0 0 0  59C                  PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC                    3416.000

05D5 0 0 0 1 3 3 E 0 00 0 000 0 0 0 0 0 0 0 0                       FULLMAR=SLEFT+DI;  COMBINE SVC POINTER & SI VECTOR              3417.000

05D6 0 0 0 0 0 0 1 0 1C 1 884 0 0 0 0 0 0 0 0                       S=T, READ, FRCWORD, SET(HIREG);  FETCH SVC CONTEXT BLOCK ADDR   3418.000

05D7 9 4 0 0 1 0 0 0 00 0 559 0 0 0 0 0 0 0 0  5D9                  IF %UPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2;                        3419.000

                                                    SVC.ERROR                                                                   3420.000

05D8 0 5 0 0 0 0 C 0 00 F 030 0 0 0 0 0 0 0 0  530                  NU=T(ZE), *GO TO CURRENT.INST.ERROR;                            3421.000

05D9 6 0 0 2 4 A F 7 02 D 040 0 0 0 0 0 0 0 0                       T=@00000400:FR(TRACE), OTHERBANK;  SET 75 MODE TRAP FLAG        3422.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SUPERVISOR CALL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

05DA 6 0 0 0 0 A B 7 02 D 100 0 0 0 0 0 0 0 0        FR(TRACE)=@10000000:T, OTHERBANK; SET TRAP FLAG                    3423.000

05DB 0 0 0 0 2 D 8 0 00 D 800 0 0 0 0 0 0 0 0        R(TRL2)=S&IO, OTHERBANK;  SAVE SVC CALL NUMBER                     3424.000

05DC 4 4 0 0 1 0 0 0 00 8 00E 0 0 0 0 0 0 0 0  5DE   IF UNBLOCK *GO TO $+2;                                             3425.000

05DD 0 6 7 0 1 0 0 0 00 0 FEB 0 0 0 0 0 0 0 0  FEB   *LINK SET.BLOCKED.M.FLAG;  GO SAVE BLOCKED CONDITION              3426.000

05DE 0 5 0 0 4 0 F 0 00 D 98E 0 0 0 0 0 0 0 0  58E   T=R(STMASK), OTHERBANK, *GO TO SVC.TRAP.ENTRY.POINT;               3427.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
SEL 32/75 CPU                                                    LOAD ADDRESS
PC  T S M A B + D R Y X PCH + B D M + B A D  ADDR

                                                                                                                    3428.000
                                        ***                                                                         3430.000
                                        *** LOAD ADDRESS (LA), (Q=0D), PRIMARY DECODE ***                           3431.000
                                        *** OPCODE = 34                                                             3432.000
                                        ***                                                                         3433.000
                                        ***                                                                         3434.000
                                        *** LOAD EFFECTIVE ADDRESS -LEA-, PRIMARY DECODE (Q=34) ***                 3435.000
                                        *** OPCODE = D0                                                             3436.000
                                        ***                                                                         3437.000

                                                                                                                    3438.000

                                        (@5E0)                                                                      3439.000

                                        LA                                                                          3440.000

                                        LEA                                                                         3441.000

05E0 3 6 0 5 1 E F 0 0C C 460 0 0 0 0 0 0 0 0  460      T=STATUS, RSTRHF, IF LATERRW *GO TO LFXTPROC;               3442.000

05E1 0 2 0 0 0 D C 0 02 0 040 0 0 0 0 0 0 0 0           NU=@04000000&T, *JUMPD;  TEST FOR EXTENDED INDEXING IN STATUS 3443.000

                                                                                                                    3444.000
                                        ***                                                                         3445.000
                                        *** LOAD ADDRESS -LA-, SECONDARY DECODE FOR OPCODE = 34, Q = 0D              3446.000
                                        ***                                                                         3447.000

                                                                                                                    3448.000

                                        LA1                                                                         3449.000

05E2 A 5 0 4 2 3 7 2 00 0 3CE 0 0 0 0 0 0 0 0  5CE      MARTX=R(X)+I0, IF %MODE75 *GO TO UNDEF.55.5XX;               3450.000

05E3 C 4 0 4 1 E F 0 02 0 009 0 0 0 0 0 0 0 0  5E9      T=@00FFFFFF, IF NALUZ *GO TO LA2; MASK = @00FFFFFF FOR 75 EXT.I 3451.000

05E4 0 4 0 5 0 D F 0 02 0 0F9 0 0 0 0 0 0 0 0  5E9      T=@FF0FFFFF&T, *GO TO LA2; MASK = @000FFFFF FOR 75 NON-EXT.I 3452.000

                                                                                                                    3453.000
                                        ***                                                                         3454.000
                                        *** LOAD EFFECTIVE ADDRESS, -LEA-, SECONDARY DECODE OF OPCODE = D0           3455.000
                                        *** Q = 34                                                                  3456.000
                                        ***                                                                         3457.000

                                                                                                                    3458.000

                                        LEA1                                                                        3459.000

05E5 2 4 0 4 2 3 7 2 00 0 307 0 0 0 0 0 0 0 0  5E7      MARIX=R(X)+I0, IF MODE75 *GO TO LEA.75; INDEX FOR INDIRECT TEST 3460.000

05E6 0 4 0 0 1 9 F 0 00 0 009 0 0 0 0 0 0 0 0  5E9      T=%S, *GO TO LEA2; MASK = @FFFFFFFF FOR 55 MODE             3461.000

                                        LEA.75                                                                      3462.000

05E7 C 4 0 4 1 E F 0 02 0 C09 0 0 0 0 0 0 0 0  5E9      T=@C0FFFFFF, IF NALUZ *GO TO LEA2; MASK = @C0FFFFFF FOR 75 EXT.I 3463.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SFL  32/75  CPU                                                    LOAD  ADDRESS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

05E8 0 0 0 5 0 D F 0 02 0 0F0 0 0 0 0 0 0 0 0          T=@FF0FFFFF&T;  MASK = @C00FFFFF FOR 75 NON-EXT.I          3464.000
                                                  ***                                                          3466.000
                                                  *** COMMON EXIT ROUTINE FOR LA AND LEA                       3467.000
                                                  ***                                                          3468.000

                                                  LA2                                                          3470.000

                                                  LEA2                                                         3471.000

05F9 A 4 0 4 2 3 7 2 00 6 C0D 0 0 0 0 0 0 0 0  5ED      MARIX=R(X)+I0, SDEST, IF %INDIR *GO TO LEA.EXIT;         3472.000

                                                  LEA.INDIRECT                                                 3473.000

05EA 0 6 7 4 2 3 7 2 00 0 FEE 0 0 0 0 0 0 0 0  FEE      MARIX=R(X)+I0, *LINK EFFECTIVE.ADDR.INDIR;               3474.000

05EB 2 4 0 0 1 0 0 0 00 0 90D 0 0 0 0 0 0 0 0  5FD      IF NCTRZ *GO TO LEA.EXIT; TEST FOR PREVIOUS MEMORY ERRORS  3475.000

05EC 0 5 0 0 0 0 C 0 00 F 030 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS  3476.000

                                                  LEA.EXIT                                                     3477.000

                                                  LA.EXIT                                                      3478.000

05FD 0 3 0 0 0 D 9 0 19 0 080 0 0 0 0 0 0 0 0          R(R)=S&T, FETCHPC, *JUMPZ;                               3479.000

                                                                                                               3480.000
                                                  ***                                                          3481.000
                                                  ***  ENABLE ARITHMETIC EXCEPTION TRAP (EAE), SECONDARY DECODE 3482.000
                                                  ***    (0=00)  OP CODE = 00                                  3483.000
                                                  ***                                                          3484.000

                                                                                                               3485.000

                                                  EAE                                                          3486.000

05EE A 5 0 0 1 0 0 0 00 0 3CE 0 0 0 0 0 0 0 0  5CE      IF %MODE75 *GO TO UNDEF.55.5XX;                          3487.000

05EF 0 0 0 0 1 0 0 0 00 0 806 0 0 0 0 0 0 0 0          SET(ENBL.AEXP);  ENABLE ARITHMETIC EXCEPTION TRAP        3488.000

05F0 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829      *GO TO FETCH.RETURN;                                     3489.000

                                                                                                               3490.000
                                                  ***                                                          3491.000
                                                  ***  DISABLE ARITHMETIC EXCEPTION TRAP (DAE), SECONDARY DECODE 3492.000
                                                  ***    (0=00)  OP CODE = 00                                  3493.000
                                                  ***                                                          3494.000

                                                                                                               3495.000

                                                  DAE                                                          3496.000

05F1 A 5 0 0 1 0 0 0 00 0 3CE 0 0 0 0 0 0 0 0  5CE      IF %MODE75 *GO TO UNDEF.55.5XX;                          3497.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                                    LOAD ADDRESS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
05F2 0 0 0 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0          RESET(ENBL.AEXP); DISABLE ARITHMETIC EXCEPTION TRAP          3498.000
05F3 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829     *GO TO FETCH.RETURN;                                         3499.000
                                                                                                                    3500.000
                                                  MP.QUNEG                                                          3501.000
05F4 0 0 0 0 1 E 6 0 00 0 007 0 0 0 0 0 0 0 0          DT=S,CLRS;                                                   3502.000
05F5 0 0 0 0 0 5 B 6 05 0 301 0 0 0 0 0 0 0 0          R(RO)=S-T,SETCAR,SETCC(D);                                   3503.000
05F6 0 6 7 0 1 0 0 0 00 D 62D 0 0 0 0 0 0 0 0  62D     OTHERBANK,*LINK FIXED.MPY.NEG.RESULT;                        3504.000
05F7 0 6 0 0 1 0 0 0 00 D 829 0 0 0 0 0 0 0 0  829     OTHERBANK,*GO TO FETCH.RETURN;                               3505.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FLOATING POINT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    3506.000
                                                                                                                    3508.000
                                                                                                                    3509.000
                                                                                                                    3510.000
                                                                                                                    3511.000
                                                                                                                    3512.000

                                                                                                                    3513.000

                                                                                                                    3514.000

                                                                                                                    3515.000

0601 0 0 0 0 4 0 F 4 1C 1 884 0 0 0 0 0 0 0 0                    T=R(R), READ, FRCWORD, SET(HIREG);                 3516.000

                                                                                                                    3517.000

0602 A 2 0 0 0 3 C 0 00 0 C09 0 0 0 0 0 0 0 0                    NU=S+T, IF %INDIR ABST, *JUMPD;                    3518.000

0603 0 0 0 4 3 3 7 1 00 D 000 0 0 0 0 0 0 0 0                    MARTX=R(DIX)+DI, OTHERBANK;                        3519.000

0604 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0                    READ, FRCWORD; FETCH OPERAND OR ANOTHER INDIRECT WORD  3520.000

0605 9 4 0 0 1 0 0 0 00 0 407 0 0 0 0 0 0 0 0  607               IF %OPNORESP:OPTIMEOUT:OPMIUER *HOP $+2; TEST PREVIOUS ERRORS  3521.000

0606 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530               NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS  3522.000

0607 5 4 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0  602               IF %EXTLW *GO TO FP1.0;                            3523.000

                                                                                                                    3524.000

0608 0 6 0 0 1 0 0 0 00 0 450 0 0 0 0 0 0 0 0  450               *GO TO EXTPROC1;                                   3525.000

                                                                                                                    3526.000

0609 0 0 0 0 0 0 0 0 00 0 301 0 0 0 0 0 0 0 0                    NOD=T,SETCC(0);                                    3527.000

                                                                                                                    3528.000

060A 0 0 1 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0                    T=S,SETCC(#);                                      3529.000

060B 0 0 0 0 2 D E 0 02 0 040 0 0 0 0 0 0 0 0                    FULLMAR=@04000000&I0;        TEST MPY OR DIV       3530.000

060C 0 0 0 5 1 E F 0 00 0 000 0 0 0 0 0 0 0 0                    T=STATUS;                                          3531.000

060D B 5 0 0 1 0 0 0 00 0 024 0 0 0 0 0 0 0 0  624               IF ALUZ *GO TO NORM.OFUF;                          3532.000

060E 0 0 0 4 2 3 7 2 00 D 000 0 0 0 0 0 0 0 0                    MARIX=R(X)+I0,OTHERBANK;                           3533.000

060F 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0                    READ,FRCWORD;                                      3534.000

0610 A 4 0 0 1 0 0 0 00 0 C04 0 0 0 0 0 0 0 0  614               IF %INDIR *GO TO TEST.MPYDIV.OFUF;                 3535.000

0611 0 6 7 0 3 0 0 0 00 0 81D 0 0 0 0 0 0 0 0  81D               NOD=DI,*LINK DELAY1;                               3536.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FLOATING POINT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0612 A 5 0 4 3 3 7 1 00 D 10F 0 0 0 0 0 0 0 0  60F        MARIX=R(DIX)+DI,OTHERBANK,IF %EXTL *GO TO $-3;                    3537.000
0613 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D        *GO TO EXTPROC1;                                                 3538.000
                                                        TEST.MPYDIV.OFUF                                                 3539.000
0614 0 0 0 0 4 0 F 4 00 D 000 0 0 0 0 0 0 0 0              T=R(R),OTHERBANK;                                            3540.000
0615 0 0 0 3 0 3 1 0 02 4 009 0 0 0 0 0 0 0 0              S=0+T,BLKCAR8,ABST;       ABS OF FXP1                        3541.000
0616 0 0 0 1 2 0 0 0 02 0 080 0 0 0 0 0 0 0 0              NOD=@00080000&IO;              TEST MPY OR DIV               3542.000
0617 0 0 0 3 3 3 F 0 02 4 008 0 0 0 0 0 0 0 0              T=0+DI,BLKCAR8,ABSDI;   ABS OF FXP2                        3543.000
0618 C 4 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  61D        IF NALUZ *GO TO MPY.OFUF;                                        3544.000
0619 0 0 0 0 0 5 F 0 00 4 000 0 0 0 0 0 0 0 0              T=S-T,BLKCAR8;                                               3545.000
061A 0 0 0 0 0 3 F 0 02 4 410 0 0 0 0 0 0 0 0              T=@41000000+T,BLKCAR8;                                       3546.000
061B A 5 0 0 0 0 0 0 00 0 822 0 0 0 0 0 0 0 0  622        NOD=T,IF %ALUNEG *GO TO OF.ROUTE;                                3547.000
061C 0 5 0 0 0 0 0 0 00 0 020 0 0 0 0 0 0 0 0  620        NOD=T,*GO TO OF.ROUTE;                                           3548.000
                                                        MPY.OFUF                                                         3549.000
061D 0 0 0 0 0 3 F 0 00 4 000 0 0 0 0 0 0 0 0              T=S+T,BLKCAR8;                                               3550.000
061E 0 0 0 0 0 3 F 0 02 4 BF0 0 0 0 0 0 0 0 0              T=@BF000000+T,BLKCAR8;                                       3551.000
061F 2 4 0 0 1 0 0 0 00 0 802 0 0 0 0 0 0 0 0  622        IF ALUNEG *GO TO S+3;                                            3552.000
                                                        OF.ROUTE                                                         3553.000
0620 A 4 0 5 1 E F 0 00 0 804 0 0 0 0 0 0 0 0  624        T=STATUS,IF %ALUNFG *GO TO NORM.OFUF;                            3554.000
0621 0 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  626        *GO TO UNDERFLOW;                                                3555.000
                                                        OF.ROUTE                                                         3556.000
0622 A 4 0 5 1 E F 0 00 0 804 0 0 0 0 0 0 0 0  624        T=STATUS,IF %ALUNFG *GO TO NORM.OFUF;                            3557.000
0623 0 4 0 0 1 0 0 0 00 0 00B 0 0 0 0 0 0 0 0  62B        *GO TO OVERFLOW;                                                 3558.000
                                                        NORM.OFUF                                                        3559.000
0624 0 4 / 0 1 D 0 0 02 0 40E 0 0 0 0 0 0 0 0  62E        NOD=@40000000&N,*LINK DUD.62X;         TEST IF OFUF.TEST         3560.000
0625 B 4 0 0 1 0 0 0 00 0 00B 0 0 0 0 0 0 0 0  62B        IF ALUZ *GO TO OVERFLOW;                                         3561.000
                                                        UNDERFLOW                                                        3562.000
0626 0 0 0 4 0 D F 0 02 0 F70 0 0 0 0 0 0 0 0              T=@F7FFFFFF&T;           SET OFUF.TEST                       3563.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FLOATING POINT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0627 0 0 0 0 0 A 1 0 02 0 400 0 0 0 0 0 0 0 0          S=@40000000:T;          SET AF AND UNDERFLOW                3564.000
                                                SET.AEXP                                                      3565.000
0628 0 0 0 0 1 0 0 0 00 0 501 0 0 0 0 0 0 0 0          SETCC(S);                                                  3566.000
0629 0 0 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0          SETAEXP;                                                   3567.000
062A 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829     *GO TO FETCH.RETURN;                                       3568.000
                                                OVERFLOW                                                      3569.000
062B 0 4 0 0 0 A 1 0 02 0 488 0 0 0 0 0 0 0 0  628     S=@48000000:T,*HOP SET.AEXP;          SET AF AND OVERFLOW  3570.000
                                                FIXED.MPY.POS.RESULT                                          3571.000
062C 0 1 0 0 1 E 9 0 00 0 001 0 0 0 0 0 0 0 0          R(R)=S,SETCC(AL),*JUMPJ;                                   3572.000
                                                FIXED.MPY.NEG.RESULT                                          3573.000
062D 0 1 0 0 3 5 9 0 06 0 001 0 0 0 0 0 0 0 0          R(R)=S-DI,USECAR,SETCC(AL),*JUMPJ;                         3574.000
                                                DUD.62X                                                       3575.000
062E 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *JUMPJ;                                                    3576.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    ADFW-SUFW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                3577.000
                                                *FP                             3578.000
                                                (@630)                          3579.000
                                                ADF                             3580.000
                                                SUF                             3581.000
0630 A 4 0 0 0 0 F 0 03 0 E05 0 0 0 0 0 0 0 0  635      T=T,FILLEXP,IF %FC1V *GO TO ADFW;   3582.000
0631 0 0 0 0 4 0 F 4 00 D 000 0 0 0 0 0 0 0 0           T=R(R),OTHERBANK;                   3583.000
0632 0 0 0 3 0 3 C 0 02 4 009 0 0 0 0 0 0 0 0           NU=0+T,BLKCAR8,ABST;                3584.000
0633 0 6 0 0 0 0 1 0 03 0 738 0 0 0 0 0 0 0 0  736      S=T,FILLEXP,*GO TO ADFU;            3585.000
                                                FPSP.FIX.N                      3586.000
0634 0 0 3 0 0 0 F 0 03 0 000 0 0 0 0 0 0 0 0           DECRN,T=T,FILLEXP;  CHANGE 80000000 TO FFF00000  3587.000
                                                ADFW                            3588.000
                                                SUFW                            3589.000
                                                **NOTE----------ADD 3 CLOCKS FOR PRIMARY AND SECONDARY DECODE  3590.000
                                                **              **S-REGISTER CONTAINS ZEROES UPON ENTRY**  3591.000
                                                **              **T-REGISTER CONTAINS R(R) FRACTION    **  3592.000
                                                **              **N-COUNTER CONTAINS EXPONENT 1 ABSOLUTE TRUE  3593.000
0635 2 4 0 0 1 0 0 0 00 2 004 0 0 0 0 0 0 0 0  634      TNIBL,IF NCTR0 *GO TO FPSP.FIX.N;   3594.000
0636 0 4 0 0 0 0 8 0 00 0 00E 0 0 0 0 0 0 0 0  63E      R(TMP0)=T,*GO TO SCALER;  CREATE FRACTION 1 GUARD DIGIT  3595.000
                                                REPLACE                         3596.000
0637 0 0 0 3 3 3 1 0 02 0 008 0 0 0 0 0 0 0 0           S=0+DI,ABSDI;                       3597.000
0638 0 0 0 0 1 5 0 0 00 4 000 0 0 0 0 0 0 0 0           NOD=S-N,BLKCAR8;                    3598.000
0639 3 4 0 0 4 0 1 0 00 2 00C 0 0 0 0 0 0 0 0  63C      S=R(TMP0),IF ALUNEGW *GO TO FPSP.FIX.DI;  3599.000
063A A 5 0 0 1 0 0 0 00 0 84F 0 0 0 0 0 0 0 0  64E      IF %ALUNEG *GO TO OPR1.LT.OPR2.GT6;  3600.000
063B 0 5 0 0 1 0 0 0 00 0 04D 0 0 0 0 0 0 0 0  64D      *GO TO OPR1.GT.OPR2.GT6;            3601.000
                                                FPSP.FIX.DI                     3602.000
063C 0 0 0 1 3 A 6 0 02 0 F00 0 0 0 0 0 0 0 0           DI=@00F00000:DI;          FORCE OPR2 TO XXF00000  3603.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    ADFW-SUFW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

063D 0 0 0 0 3 3 6 0 02 0 010 0 0 0 0 0 0 0 0          DI=@01000000+DI;          DECREMENT OPR2  EXPONENT BY 1          3604.000
                                                SCALER                                                                  3605.000
                                                **                                                                      3606.000
                                                **     N CONTAINS POSITIVE EXPONENT OPERAND 1                           3607.000
                                                **     T CONTAINS FRACTION OPERAND 1 WITH GUARD DIGIT                   3608.000
                                                **                                                                      3609.000
063E 0 0 0 0 3 0 F 0 03 0 000 0 0 0 0 0 0 0 0          T=DI,FILLEXP;             FRACTION 2 TO S                        3610.000
063F 6 0 6 0 4 0 1 0 00 2 000 0 0 0 0 0 0 0 0          S=R(TMP0),TNIBL,SCALE;    CREATE GUARD DIGIT OPERAND 2           3611.000
                                                SCALE                                                                   3612.000
                                                **     FRACTION 1 IN S-REG,R(TMP0)/FRACTION 2 IN T-REG                  3613.000
                                                **                                                                      3614.000
                                                **          OPERAND1 EXPONENT > OPERAND2 EXPONENT                       3615.000
                                                **                                                                      3616.000
0640 0 0 0 0 1 0 0 0 0A 2 080 0 0 0 0 0 0 0 0          TSFILL,TNIBR;             6-DIFFERENCE                           3617.000
0641 0 0 0 0 1 0 0 0 0A 2 080 0 0 0 0 0 0 0 0          TSFILL,TNIBR;             5-DIFFERENCE                           3618.000
0642 0 0 0 0 1 0 0 0 0A 2 080 0 0 0 0 0 0 0 0          TSFILL,TNIBR;             4-DIFFERENCE                           3619.000
0643 0 0 0 0 1 0 0 0 0A 2 080 0 0 0 0 0 0 0 0          TSFILL,TNIBR;             3-DIFFERENCE                           3620.000
0644 0 0 0 0 1 0 0 0 0A 2 080 0 0 0 0 0 0 0 0          TSFILL,TNIBR;             2-DIFFERENCE                           3621.000
0645 0 4 0 0 1 0 0 0 0A 2 08C 0 0 0 0 0 0 0 0 64C      TSFILL,TNIBR,*GO TO MERGE;        1-DIFFERENCE                   3622.000
                                                **                                                                      3623.000
                                                **          OPERAND1 EXPONENT < OPERAND2 EXPONENT                       3624.000
                                                **                                                                      3625.000
0646 0 0 0 3 1 E 1 0 0B 0 000 0 0 0 0 0 0 0 0          S=SNIBR,INCRN;                6-DIFFERENCE                       3626.000
0647 0 0 0 3 1 E 1 0 0B 0 000 0 0 0 0 0 0 0 0          S=SNIBR,INCRN;                5-DIFFERENCE                       3627.000
0648 0 0 0 3 1 E 1 0 0B 0 000 0 0 0 0 0 0 0 0          S=SNIBR,INCRN;                4-DIFFERENCE                       3628.000
0649 0 0 0 3 1 E 1 0 0B 0 000 0 0 0 0 0 0 0 0          S=SNIBR,INCRN;                3-DIFFERENCE                       3629.000
064A 0 0 0 3 1 E 1 0 0B 0 000 0 0 0 0 0 0 0 0          S=SNIBR,INCRN;                2-DIFFERENCE                       3630.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    A D F W - S U F W
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

064B 0 4 0 3 1 E 1 0 0B 0 00C 0 0 0 0 0 0 0 0  64C        S=SNIBR,INCRN,*GO TO MERGE;   1-DIFFERENCE                    3631.000
                                                MERGE                                                                   3632.000
064C 0 5 0 0 0 F 8 0 12 6 052 0 0 0 0 0 0 0 0  652        R(TMP0)=S#T, SAVESIGN, SDEST, *GO TO FAST.NORM;               3633.000
                                                OPR1.GT.OPR2.GT6                                                        3634.000
064D 0 4 0 0 4 0 F 0 00 0 F0C 0 0 0 0 0 0 0 0  64C        T=R(ZERO),*GO TO MERGE;                                       3635.000
                                                OPR1.LT.OPR2.GT6                                                        3636.000
064E 0 5 0 0 1 0 0 0 00 0 050 0 0 0 0 0 0 0 0  650        *GO TO SET.SREG.0;                                            3637.000
064F 0 5 0 0 1 0 0 0 00 0 037 0 0 0 0 0 0 0 0  637        *GO TO REPLACE;                                               3638.000
                                                SET.SREG.0                                                              3639.000
0650 0 0 0 3 3 3 C 0 02 4 008 0 0 0 0 0 0 0 0             NU=0+DI,BLKCARB,ABSDI;      EXP2 BECOMES RESULT EXP           3640.000
0651 0 5 0 0 4 0 1 0 00 0 F4C 0 0 0 0 0 0 0 0  64C        S=R(ZERO),*GO TO MERGE;                                       3641.000
                                                FAST.NORM                                                               3642.000
0652 0 0 0 3 0 3 F 0 02 0 080 0 0 0 0 0 0 0 0             T=@00000008+T;            ROUND RESULT                        3643.000
0653 1 4 0 3 1 E E 0 00 0 798 0 0 0 0 0 0 0 0  658        FULLMAR=SNIBR,IF BADSCALE *GO TO FIX.FP;                      3644.000
0654 1 4 0 0 0 0 1 0 00 0 BC6 0 0 0 0 0 0 0 0  656        S=T,IF T03SIG *GO TO FAST.FPEXIT;                             3645.000
0655 0 4 0 3 1 E E 0 00 0 00C 0 0 0 0 0 0 0 0  65C        FULLMAR=SNIBR, *GO TO DUD.65X ;                               3646.000
                                                FAST.FPEXIT                                                             3647.000
0656 0 0 0 0 1 0 9 0 19 8 081 0 0 0 0 0 0 0 0             R(R)=MAR,FETCHPC,FIXEXP,SETCC(AL);                            3648.000
0657 0 3 0 0 1 0 0 0 00 D 004 0 0 0 0 0 0 0 0             *JUMPZ,OTHERBANK,RESET(HIREG);                                3649.000
                                                FIX.FP                                                                  3650.000
0658 0 4 3 0 4 0 F 0 00 6 00D 0 0 0 0 0 0 0 0  650        DFCRN,T=R(TMP0),SDEST,*GO TO POST.NORM;                       3651.000
                                                                                                                        3652.000
                                                **                                                                      3653.000
                                                **  MULTIPLY FLOATING POINT SECONDARY DECODE                            3654.000
                                                **                                                                      3655.000
                                                                                                                        3656.000
                                                MPF                                                                     3657.000
0659 A 5 0 0 1 0 0 0 00 0 FDB 0 0 0 0 0 0 0 0  60B        IF %FC1V *GO TO MPFW;                                         3658.000
                                                MP.DOUBLE                                                               3659.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    A D F W - S U F W
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

065A 0 0 0 0 3 0 8 0 00 0 200 0 0 0 0 0 0 0 0          R(TMP2)=DI;    SAVE OPERAND 2(00-31)                    3660.000

065B 0 6 0 0 1 0 0 0 00 0 77E 0 0 0 0 0 0 0 0  77E     *GO TO MPFD;                                            3661.000

                                                  DUD.65X                                                      3662.000

065C 0 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  656     *GO TO FAST.FPEXIT ;                                    3663.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU ADFW-SUFW
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    3664.000
                                                    **                                                              3665.000
                                                    **          NEXT INSTRUCTION CREATES PATTERN FOR NORMALIZE HOP   3666.000
                                                    **                                                              3667.000
                                                    (@65D)                                                          3668.000
                                                    POST.NORM                                                       3669.000
065D 0 0 0 3 0 3 6 0 02 0 009 0 0 0 0 0 0 0 0       DI=0+T,ABST;                                                    3670.000
065E 0 0 0 3 0 3 F 0 02 0 080 0 0 0 0 0 0 0 0       T=@00000008+T;          ROUND                                   3671.000
                                                    **                                                              3672.000
                                                    **          INTERMEDIATE EXPONENT NOT ADJUSTED FOR GUARD DIGIT  3673.000
                                                    **                                                              3674.000
065F 6 0 7 0 3 0 0 0 0A 2 080 0 0 0 0 0 0 0 0       NORM,NOD=DI,TSFILL,TNIBR;                                       3675.000
                                                    ($+158-16)                                                      3676.000
                                                    NORMALIZE                                                       3677.000
                                                    R00000000                                                       3678.000
                                                    **              PROM ADDRESS = 0                                3679.000
0660 0 4 0 7 1 6 C 0 12 6 FFE 0 0 0 0 0 0 0 0 66E   NU=-1+1,SAVESIGN,SDEST,*GO TO R00XXXXXX;                        3680.000
                                                    R0000000X                                                       3681.000
                                                    **              PROM ADDRESS = 1                                3682.000
0661 0 4 0 0 4 0 C 0 12 6 FDE 0 0 0 0 0 0 0 0 66E   NU=R(ZERO),SAVESIGN,SDEST,*GO TO R00XXXXXX;                     3683.000
                                                    LXXXXXXXX.A                                                     3684.000
0662 0 4 0 0 1 0 0 0 0A 2 08A 0 0 0 0 0 0 0 0 66A   TSFILL,TNIBR,*HOP R0XXXXXXX;                                    3685.000
                                                    L0000XXXX                                                       3686.000
0663 0 4 3 2 1 E 1 0 00 0 00C 0 0 0 0 0 0 0 0 66C   DECRN,S=SNIBL,*GO TO L000XXXXX;                                 3687.000
                                                    R000000XX                                                       3688.000
                                                    **              PROM ADDRESS = 4                                3689.000
0664 0 0 0 0 4 0 F 0 00 0 F00 0 0 0 0 0 0 0 0       T=R(ZERO);                                                      3690.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          A D F W - S U F W
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0665 0 4 3 2 1 E 1 0 00 0 007 0 0 0 0 0 0 0 0  667        DECRN,S=SNIBL,*GO TO L00000XXX;                              3691.000
                                                R00000XXX                                                            3692.000
                                                **              PROM ADDRESS = 6                                     3693.000
0666 0 0 0 0 4 0 F 0 00 0 F00 0 0 0 0 0 0 0 0           T=R(ZERO);                                                   3694.000
                                                L00000XXX                                                            3695.000
0667 0 4 3 2 1 E 1 0 00 0 003 0 0 0 0 0 0 0 0  663        DECRN,S=SNIBL,*GO TO L0000XXXX;                              3696.000
                                                R0000XXXX                                                            3697.000
                                                **              PROM ADDRESS = 8                                     3698.000
0668 0 4 0 0 4 0 F 0 00 0 F03 0 0 0 0 0 0 0 0  663        T=R(ZERO),*GO TO L0000XXXX;                                  3699.000
                                                R000XXXX                                                             3700.000
                                                **              PROM ADDRESS = 9                                     3701.000
0669 0 4 0 0 4 0 F 0 00 0 F0C 0 0 0 0 0 0 0 0  66C        T=R(ZERO),*GO TO L000XXXX;                                   3702.000
                                                R0XXXXXX                                                             3703.000
                                                **              PROM ADDRESS = A                                     3704.000
066A 1 4 0 3 1 E 1 0 0B 0 BCE 0 0 0 0 0 0 0 0  66E        S=SNIBR,INCRN,IF T03SIG *GO TO R00XXXXX;                     3705.000
066B 0 4 0 0 0 0 1 0 00 0 00E 0 0 0 0 0 0 0 0  66E        S=T,*GO TO R00XXXXX;                                         3706.000
                                                L000XXXX                                                             3707.000
066C 0 4 3 2 1 E 1 0 00 0 00E 0 0 0 0 0 0 0 0  66E        DECRN,S=SNIBL,*GO TO R00XXXXX;                               3708.000
                                                RXXXXXXX                                                             3709.000
                                                **              PROM ADDRESS = D                                     3710.000
066D 0 5 0 3 1 E F 0 0B 6 074 0 0 0 0 0 0 0 0  674        T=SNIBR,INCRN,SDEST,*GO TO LXXXXXXX;                         3711.000
                                                R00XXXXX                                                             3712.000
                                                **              PROM ADDRESS = F                                     3713.000
066E 0 0 0 0 1 E E 0 00 0 000 0 0 0 0 0 0 0 0           FULLMAR=S;                                                   3714.000
                                                NORM.EXIT                                                            3715.000
066F 2 5 0 0 1 0 0 0 03 0 D0A 0 0 0 0 0 0 0 0  60A        NOD=MAR,FILLEXP,IF NCTR0 *GO TO OFUF.TEST;                   3716.000
0670 5 5 0 0 1 0 0 0 00 1 056 0 0 0 0 0 0 0 0  656        IF %SIGNSAVE *GO TO FAST.FPEXIT;                             3717.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    A D F W - S U F W
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0671 C 5 0 0 1 0 0 0 00 0 056 0 0 0 0 0 0 0 0  656          IF NALUZ *GO TO FAST.FPEXIT;                          3718.000

0672 0 0 0 0 1 E 1 0 02 0 FF0 0 0 0 0 0 0 0 0                S=@FF000000;                                          3719.000

0673 0 5 0 3 1 E E 0 08 0 05C 0 0 0 0 0 0 0 0  65C          FULLMAR=SNIBR,INCPN,*GO TO DUD.65X ;                  3720.000

                                                        LXXXXXXXX                                                  3721.000

0674 0 0 0 3 0 3 F 0 02 0 080 0 0 0 0 0 0 0 0                T=@00000008+T;                                        3722.000

0675 0 5 0 0 1 0 0 0 00 0 062 0 0 0 0 0 0 0 0  662          *GO TO LXXXXXXXX.A;                                   3723.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

SEL 32/75 CPU FIXED MPY-DIV

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    3724.000
                                                    **                                                              3725.000
                                                    **          **MULTIPLY REGISTER BY REGISTER Q=40                 3726.000
                                                    **                                                              3727.000
                                                    MPRDVR                                                          3728.000
0676 3 6 0 0 1 0 0 0 00 C 460 0 0 0 0 0 0 0 0  460      IF LATERRW *GO TO LEXTPROC;                                    3729.000
0677 1 4 0 0 4 0 F 6 00 0 BD7 0 0 0 0 0 0 0 0  677      T=R(RO),IF BIBUSY *GO TO $;                                    3730.000
0678 0 2 0 0 4 0 6 5 00 5 705 0 0 0 0 0 0 0 0           DI=R(S),TOGRHF, RESET(FLAG), *JUMPD;                           3731.000
                                                    **                                                              3732.000
                                                    **          **MULTIPLY BY MEMORY Q=C0                           3733.000
                                                    **                                                              3734.000
                                                    MPM                                                             3735.000
0679 0 0 0 0 4 0 F 6 00 0 705 0 0 0 0 0 0 0 0           T=R(RO), RESET(FLAG);                                          3736.000
067A 5 5 0 0 3 0 0 0 00 E 09E 0 0 0 0 0 0 0 0  69E      WOD=DI,IF %HWORD *GO TO FIXED.MPY;                             3737.000
067B 0 5 0 0 3 0 6 0 00 F 09E 0 0 0 0 0 0 0 0  69E      DI=DI(SE),*GO TO FIXED.MPY;                                    3738.000
                                                    **                                                              3739.000
                                                    **          **MULTIPLY IMMEDIATE Q=C8                           3740.000
                                                    **                                                              3741.000
                                                    MPI                                                             3742.000
067C 0 0 0 0 0 D 1 0 02 F 705 0 0 0 0 0 0 0 0           S=@70000000&T(ZE), RESET(FLAG); CLEAR S AND RESET FLAG F/F.    3743.000
067D 1 4 0 0 4 0 F 6 0C 0 BDD 0 0 0 0 0 0 0 0  67D      T=R(RO),RSTRHF,IF BIBUSY *GO TO $;                             3744.000
067E 0 5 0 0 2 0 6 0 00 E 09E 0 0 0 0 0 0 0 0  69E      DI=IO(SE),*GO TO FIXED.MPY;                                    3745.000
                                                    **                                                              3746.000
                                                    **          **DIVIDE MEMORY**                                   3747.000
                                                    **                                                              3748.000
                                                    DVM                                                             3749.000
067F 0 0 0 0 4 0 F 6 00 0 000 0 0 0 0 0 0 0 0           T=R(RO);                                                       3750.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
F I X E D M P Y - D I V

```
SEL 32/75 CPU
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0680 5 4 0 0 3 0 0 0 00 E 004 0 0 0 0 0 0 0 0  684        NOD=DI,IF %HWORD *GO TO FIXED.DIV;                    3751.000
0681 0 4 0 0 3 0 6 0 00 E 004 0 0 0 0 0 0 0 0  684        DI=DI(SE),*GO TO FIXED.DIV;                           3752.000
                                                                                                                3753.000
                                                **                                                              3754.000
                                                **              **DIVIDF IMMEDIATE Q=C8                         3755.000
                                                **                                                              3756.000
                                                DVI
0682 1 4 0 0 4 0 F 6 0C 0 BD2 0 0 0 0 0 0 0 0  682        T=R(RO),RSTRHF,IF BIBUSY *GO TO $;                    3757.000
0683 0 4 0 0 2 0 6 0 00 E 004 0 0 0 0 0 0 0 0  684        DI=T0(SE),*GO TO FIXED.DIV;                           3758.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SFL 32/75 CPU FIXED DIVIDE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    3759.000
                                                FIXED.DIV                                                           3760.000
                                                DVR                                                                 3761.000
0684 0 0 0 4 3 8 0 4 12 0 007 0 0 0 0 0 0 0 0       NOD=R(R)!DI,SAVESIGN,CLRS;                                      3762.000
0685 0 0 0 0 3 3 8 0 00 0 408 0 0 0 0 0 0 0 0       R(TMP4)=S+DI,ABSDI;     SAVE DIVISOR                            3763.000
0686 0 0 0 0 1 E C 0 02 D 1E7 0 0 0 0 0 0 0 0       NU=@1E000000,OTHERBANK,CLRS;   LOAD DIVIDE REDUCTION COUNT      3764.000
0687 0 0 0 0 4 0 F 4 00 0 000 0 0 0 0 0 0 0 0       T=R(R);                         LOAD DIVIDEND(00-31)            3765.000
0688 0 0 0 0 4 3 6 6 05 0 009 0 0 0 0 0 0 0 0       DI=S+R(RO),SETCAR,ABST;       ABS-TRUE DVD(32-63)               3766.000
0689 0 0 0 0 0 3 1 0 06 0 009 0 0 0 0 0 0 0 0       S=S+T,USECAR,ABST;                        ABS-TRUE DVD(00-31)     3767.000
068A 0 0 0 0 4 0 F 0 00 D 400 0 0 0 0 0 0 0 0       T=R(TMP4),OTHERBANK;     RELOAD DIVISOR                         3768.000
068B 0 0 0 0 0 5 0 0 00 0 804 0 0 0 0 0 0 0 0       NOD=S-T,SET(HIREG);                TEST DIVISOR < DIVIDEND      3769.000
068C 0 0 0 1 0 5 1 0 00 3 030 0 0 0 0 0 0 0 0       S=SLEFT-T,SHIFTDI(SLCD);        1ST REDUCTION SUBRTACT          3770.000
068D A 5 0 0 1 E 0 0 00 0 898 0 0 0 0 0 0 0 0  698  NOD=S,IF %ALUNEG *GO TO FIXDV.AE;                             3771.000
068E 6 0 5 1 0 3 1 0 00 3 03A 0 0 0 0 0 0 0 0       REPEAT,S=SLEFT+T,SHIFTDI(SLCD),DIVIDE;                          3772.000
068F 0 0 0 0 1 E 0 0 00 3 030 0 0 0 0 0 0 0 0       NOD=S,SHIFTDI(SLCD);                                            3773.000
0690 A 4 0 0 1 0 0 0 00 0 802 0 0 0 0 0 0 0 0  692  IF %ALUNEG *GO TO $+2;                                       3774.000
                                                **              IF INTERMEDIATE DIVIDEND NEGATIVE ADD IN DIVISOR    3775.000
                                                **              TO CORRECT REMAINDER                                3776.000
0691 0 0 0 0 0 3 1 0 00 0 000 0 0 0 0 0 0 0 0       S=S+T;            CORRECT REMAINDER                             3777.000
0692 0 0 0 0 3 1 6 0 00 0 000 0 0 0 0 0 0 0 0       DI=%DI;                                                         3778.000
0693 3 4 0 0 1 0 0 0 00 4 00B 0 0 0 0 0 0 0 0  698  IF BMUX00 *GO TO FIXDV.AE;                                   3779.000
0694 0 0 0 0 4 0 0 4 00 D 004 0 0 0 0 0 0 0 0       NOD=R(R),OTHERBANK,RESET(HIREG);                                3780.000
0695 3 4 0 0 1 E 9 0 00 4 007 0 0 0 0 0 0 0 0  697  R(R)=S,IF BMUX00 *GO TO $+2;                                  3781.000
0696 0 0 0 3 0 5 9 0 02 0 000 0 0 0 0 0 0 0 0       R(R)=0-T;                                                       3782.000
0697 5 4 0 0 3 0 B 6 00 1 009 0 0 0 0 0 0 0 0  699  R(RO)=DI,IF %SIGNSAVE *GO TO $+2;                            3783.000
0698 0 0 0 3 3 5 B 6 02 0 000 0 0 0 0 0 0 0 0       R(RO)=0-DI;                                                     3784.000
0699 0 0 0 0 0 0 0 0 00 0 001 0 0 0 0 0 0 0 0       NOD=T,SETCC(AL);                                                3785.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FIXED DIVIDE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
069A 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829        *GO TO FETCH.RETURN;                                    3786.000
                                                      FIXDV.AE                                                    3787.000
069B 0 0 0 0 3 0 F 0 00 0 001 0 0 0 0 0 0 0 0              T=DT,SETCC(AL);                                         3788.000
069C 0 0 0 0 1 E F 0 02 0 400 0 0 0 0 0 0 0 0              T=@40000000;                                            3789.000
069D 0 5 0 5 0 A 1 0 00 0 028 0 0 0 0 0 0 0 0  628         S=STATUS:T,*GO TO SET.AEXP;                             3790.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    FIXED MULTIPLY
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                    3791.000
                                        FIXED.MPY                                   3792.000
                                        MPR                                         3793.000
069E 0 0 0 0 0 3 8 0 00 6 709 0 0 0 0 0 0 0 0           R(TBL1)=S+T,SDEST,ABST;     3794.000
069F 0 0 0 0 0 D D 0 02 D F80 0 0 0 0 0 0 0 0           NL=@F8000000&T,OTHERBANK;   3795.000
06A0 0 0 0 0 0 3 8 0 00 0 804 0 0 0 0 0 0 0 0           R(TBL2)=S+T,SET(HIREG);     3796.000
06A1 B 5 0 0 0 3 8 0 00 D 982 0 0 0 0 0 0 0 0  6B2      R(TBL3)=S+T,OTHERBANK,IF ALUZ *GO TO MULTIPLY;  3797.000
                                        **              **MULTIPLIER BITS 0-3 NOT ZEROES  3798.000
                                        **              **CANNOT USE IT AS MULTIPLIER     3799.000
                                        **              **SO TEST MULTIPLICAND BITS 0-3 TO FIND  3800.000
                                        **              **OUT IF TIMES TABLE WILL OFUF.TEST  3801.000
06A2 0 5 7 0 1 E 8 0 00 0 02E 0 0 0 0 0 0 0 0  62E      R(TMP0)=S,*LINK DUD.62X;    SAVE OPERAND 1  3802.000
06A3 0 0 0 0 4 0 1 6 00 D 000 0 0 0 0 0 0 0 0           S=R(RO),OTHERBANK;          3803.000
06A4 0 0 0 0 3 0 B 6 00 0 000 0 0 0 0 0 0 0 0           R(RO)=DI;                   3804.000
06A5 0 0 0 0 1 E 6 0 00 D 007 0 0 0 0 0 0 0 0           DI=S,OTHERBANK,CLRS;        3805.000
06A6 0 0 0 0 0 3 8 0 12 6 709 0 0 0 0 0 0 0 0           R(TBL1)=S+T,ABST,SDEST,SAVESIGN;  3806.000
06A7 0 0 0 0 0 D 0 0 02 0 F80 0 0 0 0 0 0 0 0           NOD=@F8000000&T;            3807.000
06A8 0 0 0 0 0 3 8 0 00 0 800 0 0 0 0 0 0 0 0           R(TBL2)=S+T;                3808.000
06A9 B 5 0 0 0 3 8 0 00 0 982 0 0 0 0 0 0 0 0  6B2      R(TBL3)=S+T,IF ALUZ *GO TO MULTIPLY;  3809.000
06AA 0 6 7 0 1 0 0 0 00 0 50E 0 0 0 0 0 0 0 0  50E      *LINK TEST.FULL.SCALE.NEGATIVE;  3810.000
06AB 0 0 0 0 1 E C 0 02 2 880 0 0 0 0 0 0 0 0           INTR,NU=@80000000;          3811.000
06AC 0 0 0 0 0 D F 0 02 0 0F0 0 0 0 0 0 0 0 0           T=@0F000000&T;              3812.000
06AD 0 0 0 0 0 0 8 0 00 0 400 0 0 0 0 0 0 0 0           R(TMP4)=T;                  3813.000
06AE 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0           T=0;                        3814.000
                                        MP.USEDI                                    3815.000
06AF 0 0 0 0 0 3 8 0 00 0 700 0 0 0 0 0 0 0 0           R(TBL1)=S+T;                3816.000
06B0 0 0 0 0 0 3 8 0 00 0 800 0 0 0 0 0 0 0 0           R(TBL2)=S+T;                3817.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          F I X E D  M U L T I P L Y
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

06B1 0 4 0 0 0 3 8 0 00 0 902 0 0 0 0 0 0 0 0  6B2        R(TBL3)=S+T,*GO TO MULTIPLY;                                    3818.000

                                               MULTIPLY                                                                  3819.000

06B2 0 6 7 0 1 0 0 0 00 0 7CB 0 0 0 0 0 0 0 0  7CB        *LINK MPY.TAB+2;                                                3820.000

06B3 0 0 0 3 3 3 F 0 02 0 008 0 0 0 0 0 0 0 0             T=0+DI,ABSDI;                                                   3821.000

06B4 0 6 7 4 3 8 0 6 12 D 7D8 0 0 0 0 0 0 0 0  7D8        NOD=R(RO)!DI,SAVESIGN,OTHERBANK,*LINK MP.CYCLES;                 3822.000

06B5 0 0 0 3 1 E 1 0 00 2 080 0 0 0 0 0 0 0 0             S=SNIBR,TNIBR;                                                  3823.000

06B6 2 4 7 0 1 0 0 0 00 0 D0D 0 0 0 0 0 0 0 0  6BD        IF NCTR0 *LINK MP.CORRECTION;                                   3824.000

06B7 3 6 0 0 1 0 0 0 00 1 5F4 0 0 0 0 0 0 0 0  5F4        IF SIGNSAVE *GO TO MP.GOINEG;                                   3825.000

06B8 0 0 0 0 0 0 B 6 00 0 301 0 0 0 0 0 0 0 0             R(RO)=T,SETCC(D);                                               3826.000

06B9 0 5 7 0 1 0 0 0 00 0 02C 0 0 0 0 0 0 0 0  62C        OTHERBANK,*LINK FIXED.MPY.POS.RESULT;                           3827.000

06BA 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829        OTHERBANK,*GO TO FETCH.RETURN;                                  3828.000

                                               ($+2)                                                                     3829.000

                                               MP.CORRECTION                                                             3830.000

06BD 0 0 0 0 4 0 C 0 00 0 400 0 0 0 0 0 0 0 0             NU=R(TMP4);                                                     3831.000

06BE 0 0 0 0 4 0 6 0 00 0 F00 0 0 0 0 0 0 0 0             DI=R(ZERO);                                                     3832.000

06BF 9 4 0 0 1 0 0 0 00 0 DE1 0 0 0 0 0 0 0 0  6C1        IF %FLAG *HOP $+2; TEST FOR NOT MAX FULL SCALE MULTIPLIER        3833.000

06C0 0 0 0 2 1 E 1 0 00 2 705 0 0 0 0 0 0 0 0             S=SNIBL, TNIBL, RESET(FLAG); SET CORRECTION FACTOR              3834.000

06C1 2 1 0 2 1 E 1 0 00 2 900 0 0 0 0 0 0 0 0             S=SNIBL,TNIBL,IF NCTR7 *JUMPJ;     MULTIPLY RESULT BY 16        3835.000

06C2 0 0 3 4 0 3 F 0 05 0 000 0 0 0 0 0 0 0 0             T=R(TMP0)+T,DECRN,SETCAR;                                       3836.000

06C3 A 4 0 0 3 3 1 0 06 0 902 0 0 0 0 0 0 0 0  6C2        S=S+DI,USECAR,IF %NCTRZ *GO TO $-1;                             3837.000

06C4 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *JUMPJ;                                                         3838.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    DVFW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    3839.000
                                                    DVF                                                             3840.000
06C5 2 5 0 0 0 0 0 0 00 0 EEB 0 0 0 0 0 0 0 0  6EB      NOD=T,IF FC1V *GO TO DVFD;                                 3841.000
                                                    DVFW                                                            3842.000
                                                    **NOTE----------ADD 3 CLOCKS FOR PRIMARY AND SECONDARY DECODE   3843.000
                                                    **            **S-REGISTER CONTAINS ZEROES UPON ENTRY**         3844.000
                                                    **            **T-REGISTER CONTAINS R(R) CONTENTS    **         3845.000
06C6 0 0 0 0 0 3 8 0 03 0 109 0 0 0 0 0 0 0 0       R(TMP1)=S+T,FILLEXP,ABST;                                   3846.000
06C7 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0       R(TMP4)=N;              SAVE EXPONENT 1                     3847.000
06C8 0 0 0 3 3 3 C 0 02 4 008 0 0 0 0 0 0 0 0       NU=0+DI,BLKCAR8,ABSDI;          ABS EXP2                    3848.000
06C9 0 0 0 4 3 8 0 4 12 D 000 0 0 0 0 0 0 0 0       NOD=R(R)!DI,SAVESIGN,OTHERBANK;                             3849.000
06CA 0 0 0 4 1 5 C 0 00 4 400 0 0 0 0 0 0 0 0       NU=R(TMP4)-N,BLKCAR8;           EXP1 - EXP2                 3850.000
06CB 0 0 0 0 3 3 6 0 03 0 008 0 0 0 0 0 0 0 0       DI=S+DI,FILLEXP,ABSDI;          ABS TRUE OPR2(00-31)        3851.000
06CC 0 0 0 0 1 3 C 0 02 4 410 0 0 0 0 0 0 0 0       NU=@41000000+N,BLKCAR8;         ADJUST FOR BIAS             3852.000
06CD 8 0 0 0 1 E 0 0 02 0 87D 0 0 0 0 0 0 0 0       NOD=@87000000,IF ALUZ CLONU;                                3853.000
06CE 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0       R(TMP4)=N;                      SAVE INTERMEDIATE EXPONENT  3854.000
06CF 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0       T=DI;                                                       3855.000
06D0 0 0 0 0 1 E C 0 02 0 180 0 0 0 0 0 0 0 0       NU=@18000000;           ITERATION COUNT                     3856.000
06D1 0 0 0 0 4 0 1 0 00 2 100 0 0 0 0 0 0 0 0       S=R(TMP1),TNIBL;                SHIFT DIVIDEND              3857.000
06D2 0 0 0 3 1 E 6 0 02 0 0F0 0 0 0 0 0 0 0 0       DI=@0000000F;                                               3858.000
06D3 0 0 0 1 0 5 1 0 00 0 000 0 0 0 0 0 0 0 0       S=SLEFT-T;                                                  3859.000
                                                    **                                                          3861.000
                                                    **                                                          3862.000
                                                    **          DI=INTERMEDIATE QUOTIENT (1'S COMPL FORM)       3863.000
                                                    **          S=DIVIDEND (ABSOLUTE FORM)                      3864.000
                                                    **          T=DIVISOR (ABSOLUTE FORM)                       3865.000
                                                    **                                                          3866.000
                                                    **        28-REDUCTION CYCLES                               3867.000
                                                    **            S BIT 0 CARRY TO DI BIT 31                    3868.000
                                                    **            QUOTIENT IN DI IS ONES COMPLEMENT FORM        3869.000
                                                    **                AS FFRRRRRR + SIGN FOR ROUND              3870.000
                                                    **            DIVISOR AND DIVIDEND IN ABSOLUTE TRUE FORM    3871.000
                                                    **            ADD IF ALU SIGN - FROM PRIOR OPERATION        3872.000
                                                    **            SUB IF ALU SIGN + FROM PRIOR OPERATION        3873.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    DVFW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                        **                                                                  3874.000
06D4 6 0 5 1 0 3 1 0 00 3 03A 0 0 0 0 0 0 0 0          S=SLEFT+T,SHIFTDI(SLCD),REPEAT,DIVIDE;              3876.000
06D5 0 0 0 0 4 0 C 0 00 0 400 0 0 0 0 0 0 0 0          NII=R(TMP4);            LOAD INTERMEDIATE EXPONENT  3877.000
06D6 0 0 0 0 3 1 F 0 00 6 000 0 0 0 0 0 0 0 0          T=%DI,SDEST;                                        3878.000
06D7 5 5 0 0 1 E 0 0 00 1 05D 0 0 0 0 0 0 0 0  65D     NOD=S,IF %SIGNSAVF *GO TO POST.NORM;                3879.000
06D8 0 0 0 3 0 5 F 0 02 6 000 0 0 0 0 0 0 0 0          T=0-T,SDEST;                                        3880.000
06D9 0 5 0 0 1 E 0 0 00 0 05D 0 0 0 0 0 0 0 0  65D     NOD=S,*GO TO POST.NORM;                             3881.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    MPFW
PC   TSMAB+DR  YX PCH+BDM+BAD  ADDR

                                                                                                    3882.000
                                        (@6D8)                                                      3883.000
                                        MPFW                                                        3884.000
                                        **NOTE----------ADD 3 CLOCKS FOR PRIMARY AND SECONDARY DECODE 3885.000
                                        **              **S-REGISTER CONTAINS ZEROES UPON ENTRY**   3886.000
                                        **              **T-REGISTER CONTAINS R(R) CONTENTS    **   3887.000
                                        **              **N-COUNTER CONTAINS EXPONENT1 ABSOLUTE TRUE** 3888.000
06D8 0 0 0 0 0 3 8 0 03 6 709 0 0 0 0 0 0 0 0          R(TBL1)=S+T,FILLEXP,SDEST,ABST;             3889.000
06DC 0 6 7 0 1 0 0 0 00 0 7C9 0 0 0 0 0 0 0 0  7C9     *LINK MPY.TAB;                              3890.000
06DD B 5 0 0 1 0 8 0 00 0 460 0 0 0 0 0 0 0 0  660     R(TMP4)=N,IF ALUZ *GO TO R00000000;    SAVE EXPONENT 1  3891.000
                                        **NOTE----------ADD 8 CLOCKS FOR TABLE BUILD COMPLETION     3892.000
06DE 0 0 0 3 3 3 F 0 02 0 008 0 0 0 0 0 0 0 0          T=0+DI,ABSDI;            ABS TRUE EXP 2     3893.000
06DF 0 0 0 4 0 3 C 0 00 4 407 0 0 0 0 0 0 0 0          NOD=R(TMP4)+T,BLKCARB,CLRS;          EXP1 + EXP2  3894.000
06E0 0 0 0 0 0 1 3 C 0 02 4 BF0 0 0 0 0 0 0 0 0        NOD=@BF000000+N,BLKCARB;         ADJUST FOR BIAS  3895.000
06E1 0 0 0 0 3 3 F 0 03 0 008 0 0 0 0 0 0 0 0          T=S+DI,FILLEXP,ABSDI;            FRACTION 2      3896.000
06E2 0 0 0 0 0 0 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=T;                                           3897.000
06F3 B 5 0 4 3 8 0 4 12 D 060 0 0 0 0 0 0 0 0  660     NOD=R(R)!DI,SAVESIGN,OTHERBANK,IF ALU7 *GO TO R00000000; 3898.000
06E4 2 4 0 0 1 0 0 0 00 0 706 0 0 0 0 0 0 0 0  6E6     IF ALU4-7Z *GO TO $+2;                           3899.000
06E5 0 0 0 0 1 0 0 0 00 2 080 0 0 0 0 0 0 0 0          TNIBR;                           ADJUST FOR NEG FULLSCALE 3900.000
06E6 0 6 7 0 0 0 0 0 00 0 7D7 0 0 0 0 0 0 0 0  7D7     NOD=T,*LINK FPSW.MPY;                            3901.000
                                        **NOTE----------ADD 9 CLOCKS FOR MULTIPLY IF NO CARRY       3902.000
                                        **                13 CLOCKS FOR MULTIPLY IF CARRY            3903.000
06E7 5 4 0 0 1 E 6 0 00 1 00A 0 0 0 0 0 0 0 0  6FA     DI=S,IF %SIGNSAVE *GO TO $+3;                    3904.000
06E8 0 0 0 4 0 5 0 0 05 0 F00 0 0 0 0 0 0 0 0          NOD=R(ZERO)-T,SETCAR;                            3905.000
06E9 0 0 0 4 3 5 F 0 06 6 F00 0 0 0 0 0 0 0 0          T=R(ZERO)-DI,USECAR,SDEST;                       3906.000
06EA 0 5 0 0 1 E F 0 00 0 05D 0 0 0 0 0 0 0 0  65D     T=S,*GO TO POST.NORM;                            3907.000
```

```
                    * S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 8 5  C P U                                                                          D V F D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                        3908.000
                                                    DVFD                                                                3909.000
06EB 0 5 7 0 3 0 8 0 00 0 22E 0 0 0 0 0 0 0 0  62E      R(TMP2)=DI,*LINK DUD.62X;          SAVE OPR2(00-31)         3910.000
06EC 0 0 0 7 4 D F 6 00 D F00 0 0 0 0 0 0 0 0           T=@FFFFFFF0&R(R0),OTHERBANK;    REMOVE LS-DIGIT             3911.000
06ED 0 0 0 4 3 8 0 4 12 D 000 0 0 0 0 0 0 0 0           NOD=R(R)!DI,SAVESIGN,OTHERBANK;                             3912.000
06EE 0 6 7 0 4 0 6 4 00 D 72F 0 0 0 0 0 0 0 0  72F      DI=R(R),OTHERBANK,*LINK ABS.OPR;                            3913.000
06EF 0 0 0 3 1 E 1 0 1C 2 082 0 0 0 0 0 0 0 0           S=SNIBR,TNIBR,READ,FORCEZ;              READ OPR2(32-63)    3914.000
06F0 0 0 0 0 0 0 8 0 00 0 000 0 0 0 0 0 0 0 0           R(TMP0)=T;              DIVIDEND(32-63)                     3915.000
06F1 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0           R(TMP4)=N;              SAVE EXP1                           3916.000
06F2 0 0 0 0 1 E 8 0 00 0 100 0 0 0 0 0 0 0 0           R(TMP1)=S;              DIVIDEND(00-31)                     3917.000
06F3 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0           T=DI;                                                       3918.000
06F4 0 0 0 0 4 0 6 0 03 0 207 0 0 0 0 0 0 0 0           DI=R(TMP2),FILLEXP,CLRS;                                    3919.000
06F5 0 0 0 0 0 3 F 0 05 0 008 0 0 0 0 0 0 0 0           T=S+T,SETCAR,ABSDT;             ABS TRUE OPR2(32-63)        3920.000
06F6 0 0 0 0 4 3 C 0 06 0 208 0 0 0 0 0 0 0 0           NU=S+R(TMP2),USECAR,ABSDI;      ABS OF EXP2                 3921.000
06F7 0 0 0 4 1 5 C 0 00 4 400 0 0 0 0 0 0 0 0           NU=R(TMP4)-N,BLKCAR8;           EXP1 - EXP2                 3922.000
06F8 0 0 0 0 1 3 C 0 02 4 410 0 0 0 0 0 0 0 0           NU=@41000000+N,BLKCAR8;         ADJ EXP BIAS                3923.000
06F9 0 0 0 0 3 3 6 0 06 6 008 0 0 0 0 0 0 0 0           DI=S+DI,USECAR,SDEST,ABSDI;      ABS TRUE OPR2(00-31)       3924.000
06FA 0 0 0 0 0 A 0 0 00 0 000 0 0 0 0 0 0 0 0           NOD=S:T;                        OPR ZERO TEST               3925.000
06FB 2 4 0 0 0 A 0 0 00 0 700 0 0 0 0 0 0 0 0  6FD      NOD=S:T,IF ALU4-77 *GO TO $+2;                              3926.000
06FC 0 0 0 3 1 E 6 0 00 2 080 0 0 0 0 0 0 0 0           DI=SNIBR,TNIBR;                  ADJ FULLSCALE NEG          3927.000
06FD 8 0 0 0 1 E 0 0 02 0 8FD 0 0 0 0 0 0 0 0           NOD=@8F000000,IF ALUZ CLDNU;    FORCE OVERFLOW              3928.000
06FE 0 0 0 0 0 0 8 0 0B 0 E07 0 0 0 0 0 0 0 0           R(TBL8)=T,CLRS,INCRN;                SAVE + DVR(32-63)      3929.000
06FF 0 0 0 0 0 5 8 0 05 0 F00 0 0 0 0 0 0 0 0           R(TBL8)=S-T,SETCAR;             SAVE - DVR(32-63)           3930.000
0700 0 0 0 0 3 5 8 0 06 D D00 0 0 0 0 0 0 0 0           R(TBL7)=S-DI,USECAR,OTHERBANK;  SAVE - DVR(00-31)           3931.000
0701 0 0 0 0 3 0 8 0 00 D D00 0 0 0 0 0 0 0 0           R(TBL7)=DI,OTHERBANK;           SAVE + DVR(00-31)           3932.000
                                                  FPDW.DIV0                                                         3933.000
0702 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0           R(TMP4)=N;              SAVE INTERMEDIATE EXPONENT          3934.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    DVFD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0703 0 0 0 0 1 E F 0 02 0 030 0 0 0 0 0 0 0 0          T=@03000000;                                          3935.000

0704 0 0 0 0 4 0 6 0 00 0 000 0 0 0 0 0 0 0 0          DI=R(TMP0);            RELOAD DIVIDEND(32-63)         3936.000

0705 0 0 0 0 4 0 1 0 00 0 100 0 0 0 0 0 0 0 0          S=R(TMP1);             RELOAD DIVIDEND(00-31)         3937.000

0706 0 0 0 0 0 0 8 0 00 0 300 0 0 0 0 0 0 0 0          R(TMP3)=T;             SAVE ITERATION PASS COUNT      3938.000

0707 0 0 0 0 1 E C 0 02 0 070 0 0 0 0 0 0 0 0          N=@07000000;           1ST ITERATION COUNT            3939.000

                                                  **          FIRST REDUCTION ALWAYS SUBTRACT                3940.000

0708 0 0 0 0 1 0 0 0 01 3 020 0 0 0 0 0 0 0 0          SHIFTD(SLLD);                                         3941.000

0709 0 0 0 4 3 3 6 0 05 0 E00 0 0 0 0 0 0 0 0          DT=R(TBL8)+DI,SETCAR,OTHERBANK;                       3942.000

070A 0 4 0 0 4 3 1 0 06 D D0D 0 0 0 0 0 0 0 0  70D     S=S+R(TBL7),USECAR,OTHERBANK,*GO TO DP.DIV+2;          3943.000
                                                  **                                                        3945.000
                                                  **          DIVIDE REDUCTION                              3946.000
                                                  **                  S AND DT CONTAIN DIVIDEND             3947.000
                                                  **                  R(TBL7) AND R(TBL8) CONTAIN + DIVISOR 3948.000
                                                  **                  R(TBL7) AND R(TBL8) CONTAIN - DIVISOR IN 3949.000
                                                  **                  LOWER REGISTER BANK                   3950.000
                                                  **                  N CONTAINS REDUCTION COUNT            3951.000
                                                  **                  QUOTIENT BITS ACCUMULATED IN SERIAL IN 3952.000
                                                  **                  SHIFT REGISTER 16-BITS AT A TIME      3953.000
                                                  **                  HARDWARE SELECTS CORRECT DIVISOR ACCORDING 3954.000
                                                  **                  TO THE SIGN OF LAST REDUCTION AS FOLLOWS 3955.000
                                                  **                          - SELECTS UPPER REGISTER BANK 3956.000
                                                  **                          + SELECTS LOWER REGISTER BANK 3957.000
                                                  **                                                        3958.000

                                                  DP.DIV                                                    3960.000

070B 6 0 4 4 3 3 6 0 05 0 E00 0 0 0 0 0 0 0 0          DT=R(TBL8)+DI,DIVLSW,SETCAR;                          3961.000

070C 6 0 3 0 4 3 1 0 06 0 D00 0 0 0 0 0 0 0 0          S=S+R(TBL7),DIVMSW,USECAR;                            3962.000

070D A 4 3 1 1 E 1 0 0F 3 928 0 0 0 0 0 0 0 0  70B     DECRN,S=SLEFT,SHIFTDI(SLLD),CLKDIV,IF %NCTRZ *GO TO $-2; 3963.000

070E 6 0 4 4 3 3 6 0 05 0 F00 0 0 0 0 0 0 0 0          DT=R(TBL8)+DI,DIVLSW,SETCAR;                          3964.000

070F 6 0 3 0 4 3 1 0 06 0 D00 0 0 0 0 0 0 0 0          S=S+R(TBL7),DIVMSW,USECAR;                            3965.000

0710 0 5 7 0 4 0 C 0 00 0 367 0 0 0 0 0 0 0 0  767     N=R(TMP3),*LINK DUD.5XX;                              3966.000

0711 2 5 3 0 1 0 0 0 00 0 920 0 0 0 0 0 0 0 0  720     DECRN,IF NCTR7 *GO TO DP.DIV.DONE;                    3967.000

0712 0 5 7 0 1 0 8 0 00 0 367 0 0 0 0 0 0 0 0  767     R(TMP3)=N,*LINK DUD.5XX;                              3968.000

0713 2 4 3 0 1 0 0 0 00 0 90C 0 0 0 0 0 0 0 0  71C     DECRN,IF NCTRZ *GO TO DP.0.1;                         3969.000

0714 2 4 0 0 1 0 0 0 00 0 90A 0 0 0 0 0 0 0 0  71A     IF NCTRZ *GO TO DP.0.2;                               3970.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    DVFD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    **                    QUOTIENT BITS(00-07)                    3971.000
0715 0 4 7 0 1 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  71E          *LINK DP.DIV.SUB;                                      3972.000
0716 0 0 0 0 0 1 E 8 0 00 0 800 0 0 0 0 0 0 0 0              R(TBL2)=S;                                             3973.000
                                                    DP.DIV.REENTRY                                              3974.000
0717 3 6 0 0 1 0 1 0 00 0 45D 0 0 0 0 0 0 0 0  450          S=MAR,IF EXTLW *GO TO EXTPROC1;                        3975.000
0718 0 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0              NU=@0F000000;                                          3976.000
0719 0 5 0 0 1 E 0 0 00 0 00D 0 0 0 0 0 0 0 0  70D          NOU=S,*GO TO DP.DIV+2;                                 3977.000
                                                    **                    QUOTIENT BITS(08-23)                    3978.000
                                                    DP.Q.2                                                      3979.000
071A 0 4 7 0 1 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  71E          *LINK DP.DIV.SUB;                                      3980.000
071B 0 4 0 0 1 E 8 0 00 0 907 0 0 0 0 0 0 0 0  717          R(TBL3)=S,*GO TO DP.DIV.REENTRY;                       3981.000
                                                    **                    QUOTIENT BITS(24-39)                    3982.000
                                                    DP.Q.1                                                      3983.000
071C 0 4 7 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  71E          *LINK DP.DIV.SUB;                                      3984.000
071D 0 4 0 0 1 E 8 0 00 0 A07 0 0 0 0 0 0 0 0  717          R(TBL4)=S,*GO TO DP.DIV.REENTRY;                       3985.000
                                                    DP.DIV.SUB                                                  3986.000
071E 0 0 0 0 1 E E 0 00 0 00D 0 0 0 0 0 0 0 0                FULLMAR=S,CLONU;        SAVE INTERMEDIATE DIVIDEND     3987.000
071F 0 1 0 0 1 0 0 0 10 6 000 0 0 0 0 0 0 0 0                SELSPARE,SDEST,*JUMPJ;  READ PARTIAL INTERMEDIATE QUOTIENT 3988.000
                                                    DP.DIV.DONE                                                 3989.000
0720 0 0 0 7 1 E F 0 00 0 FF0 0 0 0 0 0 0 0 0                T=@FFFFFFFF;                                           3990.000
0721 0 5 7 0 0 0 6 0 00 F 01F 0 0 0 0 0 0 0 0  71F          DI=T(ZE),*LINK DP.DIV.SUB+1;                           3991.000
0722 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0                T=S&DI;       QUOTIENT BITS(40-55)                      3992.000
0723 0 0 0 0 3 0 1 0 00 0 000 0 0 0 0 0 0 0 0                S=DI;                                                  3993.000
0724 0 0 0 0 5 B 1 0 00 0 A00 0 0 0 0 0 0 0 0                S=%S&R(TBL4,HWS);                                      3994.000
0725 0 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0              T=S:T;           QUOTIENT BITS(24-55)                  3995.000
0726 0 0 0 0 3 0 1 0 00 0 000 0 0 0 0 0 0 0 0                S=DI;                                                  3996.000
0727 0 0 0 0 4 0 C 0 00 0 400 0 0 0 0 0 0 0 0                NU=R(TMP4);                                            3997.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU

```
                                                                                                                                  D V F D
PC   T S M A B + D R  Y X PCH + B D M + R A D  ADDR

0728 0 0 0 0 5 B 1 0 00 0 800 0 0 0 0 0 0 0 0          S=%S&R(TBL2,HWS);                                                          3998.000

0729 0 0 0 0 4 A 6 0 00 F 900 0 0 0 0 0 0 0 0          DI=S:R(TBL3,ZE);              QUOTTENT BITS (00-23)                        3999.000

072A 0 0 0 4 3 D 1 0 02 6 000 0 0 0 0 0 0 0 0          S=@00FFFFFF&DI,SDEST;                                                      4000.000

072B 5 5 0 0 1 E 6 0 00 1 05F 0 0 0 0 0 0 0 0  75E     DI=S,IF %SIGNSAVE *GO TO FPDW.NORMALIZE;                                    4001.000

072C 0 0 0 4 0 5 F 0 05 0 F00 0 0 0 0 0 0 0 0          T=R(ZERO)-T,SETCAR;                                                        4002.000

072D 0 0 0 4 3 5 1 0 06 0 F00 0 0 0 0 0 0 0 0          S=R(ZERO)-DI,USECAR;                                                       4003.000

072E 0 5 0 0 3 0 0 0 00 0 05E 0 0 0 0 0 0 0 0  75E     NOD=DI,*GO TO FPDW.NORMALIZE;                                              4004.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                    D P   O P R   A B S - T R U E
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4005.000
                                                ABS.OPR                                                             4006.000
072F 0 0 0 0 0 3 F 0 05 0 008 0 0 0 0 0 0 0 0       T=S+T,SETCAR,ABSDI;          ABS OF OPR(32-63)                 4007.000
0730 0 0 0 0 0 3 3 C 0 06 0 008 0 0 0 0 0 0 0 0     NOD=S+DI,USECAR,ABSDI;        ABS OF EXP                        4008.000
0731 0 0 0 0 0 3 0 6 0 03 0 000 0 0 0 0 0 0 0 0     DT=DI,FILLEXP;               REMOVE EXPONENT                   4009.000
0732 0 0 0 0 0 3 3 1 0 06 0 008 0 0 0 0 0 0 0 0     S=S+DI,USECAR,ABSDI;         ABS OF OPR(00-31)                 4010.000
0733 0 0 0 0 0 A 0 0 00 0 000 0 0 0 0 0 0 0 0       NOD=S:T;                     TEST BOTH OPR ZERO                4011.000
0734 2 1 0 0 0 A 0 0 00 0 700 0 0 0 0 0 0 0 0       NOD=S:T,IF ALU4-77 *JUMPJ;                                     4012.000
0735 0 1 0 3 1 E 1 0 00 2 080 0 0 0 0 0 0 0 0       *JUMPJ,S=SNIBR,TNIBR;              SHIFT FULLSCALE NEG     4013.000
                                                *                                                               4014.000
                                                (@736)                                                          4015.000
                                                *                                                               4016.000
                                                FPDW.NORM.2                                                     4017.000
                                                *                                                               4018.000
0736 C 5 0 0 1 0 0 0 00 0 06C 0 0 0 0 0 0 0 0  76C  IF NALUZ *GOTO FPDW.NORM.3;    JUMP IF S .NE. @FFF00000    4019.000
0737 0 5 0 0 1 0 0 0 00 0 06F 0 0 0 0 0 0 0 0  76F  *GOTO FPDW.EXIT;    OMIT NORMALIZING IF = @FFF0000000000000    4020.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          A D F D - S U F D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4021.000
                                                                ADFD                                                4022.000
                                                                SUFD                                                4023.000
0738 0 0 3 0 4 0 8 6 00 0 100 0 0 0 0 0 0 0 0                       DECRN,R(TMP1)=R(R0),OTHERBANK;                   4024.000
0739 0 0 0 0 1 E 8 0 00 0 407 0 0 0 0 0 0 0 0                       R(TMP4)=S,CLRS;          FRACTION 1 BITS 00-31 IN TMP4 4025.000
073A 0 0 0 0 3 0 8 0 1C 0 082 0 0 0 0 0 0 0 0                       R(TMP0)=DI,READ,FORCE7;                          4026.000
073B 0 0 0 0 0 3 8 0 00 4 309 0 0 0 0 0 0 0 0                       R(TMP3)=S+T,BLKCAR8,ARST;                        4027.000
073C 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0                       S=T;                                             4028.000
073D 0 5 7 0 1 5 F 0 00 4 067 0 0 0 0 0 0 0 0  767                  T=S-N,BLKCAR8,*LINK DIV.5XX;  COMPUTE EXPONENT DIFFERENCE 4029.000
073E A 5 0 0 1 0 0 0 0B 0 840 0 0 0 0 0 0 0 0  740                  INCRN,IF %ALUNEG *GO TO $+2;                     4030.000
073F 0 4 0 0 1 0 1 0 00 0 00D 0 0 0 0 0 0 0 0  74D                  S=N,*GO TO FPDW.EXP1GT;                          4031.000
                                                                **                                                  4032.000
                                                                **          EXPONENT OF OPERAND 2 LARGER-SHIFT OPERAND 1 4033.000
                                                                **                                                  4034.000
0740 0 0 0 0 0 0 0 0 02 0 FF0 0 0 0 0 0 0 0 0                       NOD=@FF000000RT;                                 4035.000
0741 0 0 0 0 0 3 0 0 02 4 F20 0 0 0 0 0 0 0 0                       NOD=@F2000000+T,BLKCAR8;      TEST > 12          4036.000
0742 8 4 0 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  748                  IF ALUZ *GO TO FPDW.OPR1EQOPR2;                  4037.000
0743 2 4 0 0 0 0 C 0 00 0 805 0 0 0 0 0 0 0 0  745                  NU=T,IF ALUNEG *GO TO $+2;                       4038.000
0744 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  74C                  *GO TO FPDW.CLROP1;                              4039.000
0745 0 0 0 0 4 0 F 0 00 0 100 0 0 0 0 0 0 0 0                       T=R(TMP1);              LOAD FRACTION 1(32-63)   4040.000
0746 0 0 3 0 4 0 1 0 00 0 400 0 0 0 0 0 0 0 0                       DECRN,S=R(TMP4);        LOAD FRACTION 1(00-31)   4041.000
0747 6 0 5 3 1 E 1 0 00 2 080 0 0 0 0 0 0 0 0                       REPEAT,S=SNIBR,TNIBR;                            4042.000
                                                                FPDW.OPR1                                           4043.000
0748 0 0 0 0 4 0 C 0 00 0 300 0 0 0 0 0 0 0 0                       NU=R(TMP3);             LOAD EXPONENT 2          4044.000
0749 0 0 0 0 0 0 8 0 00 0 100 0 0 0 0 0 0 0 0                       R(TMP1)=T;              STORE ADJUSTED FRACTION 1(32-63) 4045.000
074A 0 0 0 0 1 E 8 0 00 0 400 0 0 0 0 0 0 0 0                       R(TMP4)=S;              STORE ADJUSTED FRACTION 1(00-31) 4046.000
                                                                *                                                   4047.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU ADFD-SUFD

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                FPDW.OPR1EQOPR2                                          4048.000
                                                *                                                        4049.000
074B 0 5 0 0 3 0 F 0 00 0 057 0 0 0 0 0 0 0 0  757      T=DI,*GO TO FPDW.SCALED;                        4050.000
                                                **                                                       4051.000
                                                **           EXPONENT OF OPERAND 2 DIFFERENCE LARGER THAN 13 4052.000
                                                **                                                       4053.000
                                                FPDW.CLROP1                                              4054.000
074C 0 4 0 0 4 0 F 0 00 6 F08 0 0 0 0 0 0 0 0  748      T=R(ZERO),SDEST,*GO TO FPDW.OPR1;               4055.000
                                                **                                                       4056.000
                                                **           EXPONENT OF OPERAND 1 LARGER-SHIFT OPERAND 2 4057.000
                                                **                                                       4058.000
                                                FPDW.EXP1GT                                              4059.000
074D 0 0 0 0 0 3 0 0 02 4 000 0 0 0 0 0 0 0 0           NOU=@0D000000+T,BLKCARB;    TEST > 12            4060.000
074E 0 0 0 3 0 5 C 0 02 4 000 0 0 0 0 0 0 0 0           NU=0-T,BLKCARB;          CONVERT DIFFERENCE TO POSITIVE 4061.000
074F A 4 0 0 1 0 0 0 00 0 801 0 0 0 0 0 0 0 0  751      IF %ALUNEG *GO TO $+2;                          4062.000
0750 0 4 0 0 3 E C 0 00 0 006 0 0 0 0 0 0 0 0  756      NU=S,BMUX=DI,*GO TO FPDW.CLROP2;  INTERLOCK ON MEM DRT 4063.000
0751 0 0 0 0 1 E 8 0 00 0 300 0 0 0 0 0 0 0 0           R(TMP3)=S;               SAVE EXPONENT 1         4064.000
0752 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0           T=DI;                    LOAD OPERAND 2(32-63)   4065.000
0753 0 0 0 0 4 0 1 0 03 0 000 0 0 0 0 0 0 0 0           S=R(TMP0),FILLEXP;            LOAD OPERAND  2 (00-31) 4066.000
0754 6 0 5 3 1 E 1 0 00 2 080 0 0 0 0 0 0 0 0           REPEAT,S=SNIBR,TNIBR;                           4067.000
0755 0 4 0 0 4 0 C 0 00 0 308 0 0 0 0 0 0 0 0  758      NU=R(TMP3),*GO TO FPDW.SCALED+1;                4068.000
                                                **                                                       4069.000
                                                **           EXPONENT OF OPERAND 1 DIFFERENCE LARGER THAN 13 4070.000
                                                **                                                       4071.000
                                                FPDW.CLROP2                                              4072.000
0756 0 4 0 0 4 0 F 0 00 6 F08 0 0 0 0 0 0 0 0  758      T=R(ZERO),SDEST,*GO TO FPDW.SCALED+1;  CLR FRAC 2(00-63) 4073.000
                                                FPDW.SCALED                                              4074.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S S I   3 2 / 7 5   C P U                                                          A D F D - S U F D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0757 0 0 0 0 4 0 1 0 03 0 000 0 0 0 0 0 0 0 0                  S=R(TMP0),FILLEXP;                                    4075.000

0758 0 0 0 0 1 E 6 0 00 0 000 0 0 0 0 0 0 0 0                  DI=S;                                                 4076.000

0759 0 0 0 0 0 F F 0 05 0 100 0 0 0 0 0 0 0 0                  T=R(TMP1)#T,SETCAR;                                   4077.000

075A 0 5 7 4 3 F 6 0 06 6 467 0 0 0 0 0 0 0 0  767             DI=R(TMP4)#DI,USECAR,SDFST,*LINK DUD.5XX;             4078.000
                                                       FPDW.REENTER.NORM                                           4079.000
075B 0 0 0 0 1 E 0 0 12 0 80F 0 0 0 0 0 0 0 0  75E             NOD=S,SAVESIGN,IF %ALUNEG *GO TO FPDW.NORMALIZE;       4080.000
                                                       **                                                          4081.000
                                                       **              RESULTS NEGATIVE CONVERT TO POSITIVE        4082.000
                                                       **                                                          4083.000
075C 0 0 0 0 0 5 0 0 05 0 F00 0 0 0 0 0 0 0 0                  NOD=R(ZERO)-T,SETCAR;                                 4084.000
075D 0 0 0 0 5 5 6 0 06 0 F0E 0 0 0 0 0 0 0 0  75E             DI=R(ZERO)-DI,USECAR,*GO TO FPDW.NORMALIZE;           4085.000
                                                       FPDW.NORMALIZE                                              4086.000
                                                       **                                                          4087.000
                                                       **                    S AND T CONTAINS TRUE SIGNED RESULT   4088.000
                                                       **                    DT CONTAIN ABS TRUE RESULT(00-31)     4089.000
                                                       **                                                          4090.000
0760 0 0 0 0 0 0 0 0 05 0 F00 0 0 0 0 0 0 0 0                  NOD=R(ZERO)-T,SETCAR;                                 4091.000
0761 0 0 0 0 0 0 0 0 02 0 000 0 0 0 0 0 0 0 0                  NOD=0-T,NORM;                                         4092.000
                                                       ($+15&-16)                                                  4093.000
                                                       **PROM ADDRESS 0  PATTERN=00000000                          4094.000
0762 0 0 5 0 0 1 0 0 0 00 0 075 0 0 0 0 0 0 0 0  775           *GO TO FPDW.SHIFT;                                    4095.000
                                                       **PROM ADDRESS 1  PATTERN=0000000X                          4096.000
0763 0 0 3 2 1 E 1 0 00 2 004 0 0 0 0 0 0 0 0  764             DFCRN,S=SNIBL,TNIBL,*GO TO $+3;                       4097.000
                                                       *                                                           4098.000
                                                       (@762)                                                      4099.000
                                                       *                                                           4100.000
                                                       FPDW.NORM.1                                                 4101.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    A D F D - S U F D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                    *                                                                  4102.000
0762 C 4 0 0 3 8 0 0 00 0 00C 0 0 0 0 0 0 0 0  76C                      NOD=S!DI, IF NALU7 *GOTO FPDW.NORM.3;  JUMP IF T .NE. 0     4103.000
0763 0 5 0 0 1 0 0 0 00 0 036 0 0 0 0 0 0 0 0  736                      *GOTO FPDW.NORM.2;                                         4104.000
                                                                    *                                                                  4105.000
                                                                    **PROM ADDRESS 4  PATTERN=000000XX                                 4106.000
0764 0 4 3 2 1 E 1 0 00 2 006 0 0 0 0 0 0 0 0  766                      DECRN,S=SNIBL,TNIBL,*GO TO $+2;                            4107.000
                                                                    EXTL.EXIT                                                          4108.000
0765 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D                      *GO TO EXTPROC1;                                           4109.000
                                                                    **PROM ADDRESS 6  PATTERN=00000XXX                                 4110.000
0766 0 4 3 2 1 E 1 0 00 2 008 0 0 0 0 0 0 0 0  768                      DECRN,S=SNIBL,TNIBL,*GO TO $+2;                            4111.000
                                                                    DUD.5XX                                                            4112.000
0767 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                           *JUMPJ;                                                    4113.000
                                                                    **PROM ADDRESS 8  PATTERN=0000XXXX                                 4114.000
                                                                    *                                                                  4115.000
                                                                    (@768)                                                             4116.000
                                                                    *                                                                  4117.000
0768 0 4 3 2 1 E 1 0 00 2 00C 0 0 0 0 0 0 0 0  76C                      DECRN,S=SNIBL,TNIBL, *GOTO FPDW.NORM.3;                    4118.000
                                                                    **PROM ADDRESS 9  PATTERN=000XXXXX                                 4119.000
0769 0 4 0 5 1 6 6 0 02 0 EF2 0 0 0 0 0 0 0 0  762                      DT=@FFEFFFFF+1, *GOTO FPDW.NORM.1;                          4120.000
                                                                    **PROM ADDRESS A  PATTERN=0XXXXXXX                                 4121.000
076A 0 4 0 3 1 E 1 0 0B 2 08E 0 0 0 0 0 0 0 0  76E                      S=SNIBR,INCRN,TNIBR,*HOP FPDW.DELAYED.EXIT;                4122.000
                                                                    FPDW.ZERO                                                          4123.000
076B 0 4 0 0 0 0 C 0 12 6 00E 0 0 0 0 0 0 0 0  76E                      NU=T,SAVESIGN,SDEST,*GO TO FPDW.DELAYED.EXIT;               4124.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU ADFD-SUFD

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4125.000
                                                *                                                                   4126.000
                                                (@76C)                                                              4127.000
                                                *                                                                   4128.000
                                                FPDW.NORM.3                                                         4129.000
                                                *                                                                   4130.000
076C 0 4 3 2 1 E 1 0 00 2 00E 0 0 0 0 0 0 0 0  76E      DECRN,S=SNIBL,TNIBL, *GOTO FPDW.DELAYED.EXIT;               4131.000
                                                **PROM ADDRESS D  PATTERN=XXXXXXXX                                  4132.000
076D 0 4 0 3 1 E 1 0 0B 2 08A 0 0 0 0 0 0 0 0  76A      S=SNIBR,INCRN,TNIBR,*HOP $-3;                               4133.000
                                                **PROM ADDRESS E  PATTERN=00XXXXXX                                  4134.000
                                                FPDW.DELAYED.EXIT                                                   4135.000
076E 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                       4136.000
                                                FPDW.EXIT                                                           4137.000
076F A 4 0 0 1 E E 0 00 0 D01 0 0 0 0 0 0 0 0  771      FULLMAR=S,IF %NCIR0 *GO TO $+2;                             4138.000
0770 0 6 0 0 1 0 0 0 00 0 609 0 0 0 0 0 0 0 0  609      *GO TO FPDW.OFUF.TEST;                                      4139.000
0771 0 0 0 0 0 0 B 6 00 0 301 0 0 0 0 0 0 0 0           R(R0)=T,SETCC(D);                                           4140.000
0772 0 0 0 4 1 D 0 0 02 D 000 0 0 0 0 0 0 0 0           NOD=@00FFFFFF&MAR,OTHERBANK;                                4141.000
0773 B 6 0 0 1 0 0 0 00 0 66F 0 0 0 0 0 0 0 0  66F      IF ALUZ *GO TO NORM.EXIT;                                   4142.000
0774 0 6 0 0 1 0 0 0 00 0 656 0 0 0 0 0 0 0 0  656      *GO TO FAST.FPEXIT;                                         4143.000
                                                FPDW.SHIFT                                                          4144.000
0775 C 4 3 2 1 E 1 0 00 2 007 0 0 0 0 0 0 0 0  777      DECRN,S=SNIBL,TNIBL,IF NALUZ *GO TO $+2;                    4145.000
                                                FPDW.SFT.RESULT.ZERO                                                4146.000
0776 0 5 0 0 4 0 F 0 00 0 F6B 0 0 0 0 0 0 0 0  76B      T=R(ZERO),*GO TO FPDW.ZERO;                                 4147.000
0777 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0           DECRN,S=SNIBL,TNIBL;                                        4148.000
0778 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0           DECRN,S=SNIBL,TNIBL;                                        4149.000
0779 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0           DECRN,S=SNIBL,TNIBL;                                        4150.000
077A 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0           DECRN,S=SNIBL,TNIBL;                                        4151.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
ADFD-SUFD

SEL 32/75 CPU

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

077B 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0          DECRN,S=SNIBL,TNIBL;                    4152.000

077C 0 5 0 0 1 E 6 0 00 0 05B 0 0 0 0 0 0 0 0  75B     DI=S,*GO TO FPDW.REENTER.NORM;          4153.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MPFD
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                4154.000
                                                **                                                                              4156.000
                                                **              EACH DOUBLEWORD OPERAND IS CONVERTED TO TWO 28-BIT               4157.000
                                                **              WORDS AS 0XXXXXXX AND 0XXXXXXX                                  4158.000
                                                **                      OPERAND 1 BITS 08-63 EQUALS                             4159.000
                                                **                              A1 = OPERAND 1 BITS 08-35                       4160.000
                                                **                              A2 = OPERAND 1 BITS 36-63                       4161.000
                                                **                      OPERAND 2 BITS 08-63 EQUALS                             4162.000
                                                **                              B1 = OPERAND 2 BITS 08-35                       4163.000
                                                **                              B2 = OPERAND 2 BITS 36-63                       4164.000
                                                **              FOUR SEPARATE MULTIPLICATION CYCLES ARE EXECUTED                 4165.000
                                                **              RESULTING IN 4 64-BIT PRODUCTS                                  4166.000
                                                **                      A2B2(00-63) = A2 * B2                                   4167.000
                                                **                      A2B1(00-63) = A2 * B1                                   4168.000
                                                **                      A1B2(00-63) = A1 * B2                                   4169.000
                                                **                      A1B1(00-63) = A1 * B1                                   4170.000
                                                **              RESULTS ARE COMBINED AS FOLLOWS                                 4171.000
                                                **                      A2B2                    0123456789ABCDEF                4172.000
                                                **                      A2B1 +          0123456789ABCDEF                        4173.000
                                                **                      A1B2 +          0123456789ABCDEF                        4174.000
                                                **                      A1B1 +   0123456789ABCDEF                               4175.000
                                                **                                                                              4176.000
                                                **                      RESULT = 0123456789ABCDEF                               4177.000
                                                **                                                                              4178.000

                                                **NOTE----------ADD 3 CLOCKS FOR PRIMARY AND SECONDARY DECODE                   4180.000

                                                (@77E)                                                                          4181.000

                                                MPFD                                                                            4182.000

077E 0 0 0 0 4 0 F 6 00 D 000 0 0 0 0 0 0 0 0           T=R(RO),OTHERBANK;                                                      4183.000

077F 0 0 0 4 3 8 0 4 12 D 000 0 0 0 0 0 0 0 0           NOD=R(R)!DI,SAVESIGN,OTHERBANK;                                         4184.000

0780 0 5 7 0 4 0 6 4 00 D 02F 0 0 0 0 0 0 0 0  72F      DI=R(R),OTHERBANK,*LINK ABS.OPR;  FRACTION 1(00-31)                      4185.000

0781 B 5 0 0 1 0 0 0 00 0 076 0 0 0 0 0 0 0 0  776      IF ALUZ *GO TO FPDW.SET.RESULT.ZERO;                                    4186.000

0782 0 5 7 0 1 0 0 0 00 0 0BE 0 0 0 0 0 0 0 0  7BE      *LINK MPFD.CRACK;                                                       4187.000

                                                **              **RESULT A1 IN S AS 0FFFFFFF                                    4188.000

                                                **                      A2 IN T AS 0FFFFFFF                                     4189.000

0783 0 0 0 0 0 0 6 0 00 0 000 0 0 0 0 0 0 0 0           DI=T;                                                                   4190.000

0784 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0           R(TMP4)=N;                      SAVE EXP1                               4191.000

0785 0 0 0 0 1 E 8 0 1C 0 182 0 0 0 0 0 0 0 0           R(TMP1)=S,READ,FORCFZ;            A1 TO TMP1                            4192.000

0786 0 5 7 0 3 0 8 0 00 6 7C3 0 0 0 0 0 0 0 0  7C3      R(TBL1)=DI,SDEST,*LINK FPDW.MPY.TAB;                                    4193.000

                                                **              **RESULT 2*A2 THRU 8*A2 IN TBL2 THRU TBL8                       4194.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL  32/75  CPU                                                    MPFD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0787 0 0 0 0 1 0 8 0 00 0 500 0 0 0 0 0 0 0 0          R(TMP5)=N;              SAVE FLAGS FROM FPDW.MPY.TAB          4195.000
0788 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0          T=DI;                   SHIFT OPR2(32-63)                     4196.000
0789 0 0 0 0 4 0 6 0 00 0 200 0 0 0 0 0 0 0 0          DI=R(TMP2);             LOAD OPERAND2(00-31)                  4197.000
078A 0 5 7 0 1 0 0 0 00 0 02F 0 0 0 0 0 0 0 0  72F     *LINK ABS.OPR;                                                4198.000
078B 8 5 0 0 1 0 0 0 00 0 076 0 0 0 0 0 0 0 0  776     IF ALUZ *GO TO FPDW.SET.RESULT.ZERO;                          4199.000
078C 0 5 7 0 1 0 0 0 00 0 0BE 0 0 0 0 0 0 0 0  7BE     *LINK MPFD.CRACK;                                             4200.000
078D 0 0 0 4 1 3 C 0 00 4 400 0 0 0 0 0 0 0 0          NU=R(TMP4)+N,BLKCARB;                                         4201.000
078E 0 0 0 0 1 3 C 0 02 4 BF0 0 0 0 0 0 0 0 0          NU=@BF000000+N,BLKCARB;       ADJUST BIAS             4202.000
                                                 **          **RESULT B1 IN S AS 0FFFFFFF                      4203.000
                                                 **                B2 IN T AS 0FFFFFFF                         4204.000
078F 0 0 0 0 0 0 8 0 00 0 300 0 0 0 0 0 0 0 0          R(TMP3)=T;          SAVE B2                                   4205.000
0790 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0          R(TMP4)=N;              SAVE INTERMEDIATE EXPONENT            4206.000
0791 0 0 0 0 1 E 8 0 00 0 000 0 0 0 0 0 0 0 0          R(TMP0)=S;                                                    4207.000
0792 0 5 7 2 1 E 8 0 00 0 DD6 0 0 0 0 0 0 0 0  7D6     R(TBL7)=SNIBL,*LINK CLRS.OB.DUD;                              4208.000
0793 0 0 0 0 4 0 F 0 00 6 300 0 0 0 0 0 0 0 0          T=R(TMP3),SDEST;     TLOAD REG TO CLEAR MSD OF T              4209.000
0794 0 5 7 2 1 E 8 0 00 0 ED6 0 0 0 0 0 0 0 0  7D6     R(TBL8)=SNIBL,*LINK CLRS.OB.DUD;                              4210.000
                                                 **                                                            4211.000
                                                 **          A2B2 REDUCTION                                    4212.000
                                                 **                                                            4213.000
0795 0 0 0 0 4 0 C 0 00 0 500 0 0 0 0 0 0 0 0          NU=R(TMP5);             RESTORE FLAGS                         4214.000
0796 0 5 7 0 4 0 F 0 00 0 3D8 0 0 0 0 0 0 0 0  7D8     T=R(TMP3),*LINK MP.CYCLES;                                    4215.000
0797 A 5 7 0 0 0 6 0 00 0 9D3 0 0 0 0 0 0 0 0  7D3     DI=T,IF %NCTRZ *LINK ADJ.B2;                                  4216.000
0798 0 0 0 4 0 D 0 0 02 0 0F0 0 0 0 0 0 0 0 0          NOD=@0FFFFFFF&T;                                              4217.000
0799 0 0 0 3 1 E E 0 02 0 000 0 0 0 0 0 0 0 0          FULLMAR=0;                                                    4218.000
079A C 4 0 2 1 E 8 0 00 0 60C 0 0 0 0 0 0 0 0  79C     R(TMP6)=SNIBL,IF NALU7 *HOP $+2; A2B2(04-35)                  4219.000
079B 0 0 0 3 1 E E 0 02 0 010 0 0 0 0 0 0 0 0          FULLMAR=@00000001;                                            4220.000
079C 0 0 0 0 1 E F 0 00 0 007 0 0 0 0 0 0 0 0          T=S,CLRS;                                                     4221.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    MPFD
PC   TSMAB+DR  YX PCH+BDM+BAD  ADDR

079D 0 5 7 2 1 E 8 0 00 0 5C1 0 0 0 0 0 0 0 0  7C1       R(TMP5)=SNIBL,*LINK CLRS.DUD;     A2B2(00-03)                4222.000
                                                **                                                                   4223.000
                                                **            A2B1 REDUCTION                                         4224.000
                                                **                                                                   4225.000
079E 0 5 7 0 4 0 F 0 00 0 0D8 0 0 0 0 0 0 0 0  7D8       T=R(TMP0),*LINK MP.CYCLES;                                  4226.000
                                                **            **RESULTS IN S-T = A2B1                                4227.000
079F A 5 7 0 0 0 6 0 00 0 9D0 0 0 0 0 0 0 0 0  7D0       DI=T,IF %NCTRZ *LINK ADJ.B1;                                4228.000
07A0 0 5 7 4 0 3 8 0 05 0 667 0 0 0 0 0 0 0 0  767       R(TMP6)=R(TMP6)+T,SFTCAR,*LINK DUD.5XX;                     4229.000
                                                **                  A2B1(32-63) + A2B2(04-35)                        4230.000
07A1 0 5 7 0 4 3 8 0 06 0 567 0 0 0 0 0 0 0 0  767       R(TMP5)=S+R(TMP5),USECAR,*LINK DUD.5XX;                     4231.000
                                                **            A2B2(00-03) + A2B1(00-31)                              4232.000
07A2 3 5 0 0 4 0 6 0 00 0 365 0 0 0 0 0 0 0 0  765       DI=R(TMP3),IF EXTLW *GO TO EXTL.EXIT; B2 TO DI               4233.000
07A3 0 0 0 0 4 0 1 0 00 0 100 0 0 0 0 0 0 0 0            S=R(TMP1);              LOAD A1                              4234.000
07A4 0 5 7 0 1 E 8 0 00 0 7C3 0 0 0 0 0 0 0 0  7C3       R(TBL1)=S,*LINK FPDW.MPY.TAB;       A1 TO S                  4235.000
                                                **            **RESULT 2*A1 THRU 8*A1 IN TBL2 THRU TBL8              4236.000
                                                **                                                                   4237.000
                                                **            A1B2 REDUCTION                                         4238.000
                                                **                                                                   4239.000
07A5 0 5 7 0 3 0 F 0 00 0 0D8 0 0 0 0 0 0 0 0  7D8       T=DI,*LINK MP.CYCLES;                                       4240.000
                                                **            **RESULTS IN S-T = A1B2                                4241.000
07A6 A 5 7 0 0 0 6 0 00 0 9D3 0 0 0 0 0 0 0 0  7D3       DI=T,IF %NCTR7 *LINK ADJ.B2;                                4242.000
                                                **                A2B1(32-63)+A1B2(32-63)+A2B2(04-35)                4243.000
07A7 3 5 0 4 0 3 F 0 05 0 665 0 0 0 0 0 0 0 0  765       T=R(TMP6)+T,SETCAR,IF EXTLW *GO TO EXTL.EXIT;                4244.000
07A8 0 0 0 4 0 D 0 0 02 0 0F0 0 0 0 0 0 0 0 0            NOD=@0FFFFFFF&T;                                            4245.000
07A9 0 0 0 0 4 3 6 0 06 6 500 0 0 0 0 0 0 0 0            DI=S+R(TMP5),USECAR,SDEST;                                  4246.000
                                                **                                  A2B1(00-31)+A1B2(00-31)+A2B2(00-03)  4247.000
07AA C 4 0 2 1 E 8 0 00 0 50C 0 0 0 0 0 0 0 0  7AC       R(TMP5)=SNIBL,IF NALUZ *GO TO $+2;                          4248.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    MPFD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

07AB 0 0 0 3 1 A E 0 02 0 020 0 0 0 0 0 0 0 0        FULLMAR=@00000002:MAR;                                    4249.000
07AC 0 0 0 0 1 E F 0 00 0 007 0 0 0 0 0 0 0 0        T=S,CLRS;                                                 4250.000
07AD 0 5 7 2 1 E 8 0 00 0 2C1 0 0 0 0 0 0 0 0  7C1   R(TMP2)=SN1BL,*LINK CLRS.DUD;    A1B2(00-03)+A2B1(00-03)   4251.000
                                                **                                                        4252.000
                                                **          A1b1 REDUCTION                                4253.000
                                                **                                                        4254.000
07AE 0 5 7 0 4 0 F 0 00 0 0D8 0 0 0 0 0 0 0 0  7D8   T=R(TMP0),*LINK MP.CYCLES;                            4255.000
                                                **          **RESULTS IN S-T = A1B1                       4256.000
07AF A 5 7 0 0 0 6 0 00 0 900 0 0 0 0 0 0 0 0  7D0   DI=T,IF %NCTR7 *LINK ADJ.B1;                          4257.000
07B0 0 0 0 4 0 3 F 0 05 0 500 0 0 0 0 0 0 0 0        T=R(TMP5)+T,SETCAR;                                       4258.000
                                                **          T=A1b1(32-63)+(A1B2(04-35)+A2B1(04-35)+A2B2(00-07))  4259.000
07B1 0 0 0 0 4 3 6 0 06 6 200 0 0 0 0 0 0 0 0        DI=S+R(TMP2),USECAR,SDEST;                                4260.000
                                                **          S=A1b1(00-31)+(A1B2(00-03)+A2B1(00-03))       4261.000
07B2 0 0 0 3 0 D 0 0 02 0 0F0 0 0 0 0 0 0 0 0        NOD=@0000000F&T;                                          4262.000
07B3 0 0 0 3 1 E 6 0 00 2 080 0 0 0 0 0 0 0 0        DI=SN1BR,TN1BR;                                           4263.000
07B4 C 4 0 0 4 0 C 0 00 0 406 0 0 0 0 0 0 0 0  7B6   NU=R(TMP4),IF NALUZ *GO TO $+2;                         4264.000
07B5 0 0 0 3 1 A E 0 02 0 040 0 0 0 0 0 0 0 0        FULLMAR=@00000004:MAR;                                    4265.000
07B6 0 0 0 3 1 C 0 0 02 0 070 0 0 0 0 0 0 0 0        NOD=@00000007&%MAR;                                       4266.000
07B7 5 5 0 0 3 0 1 0 08 1 05F 0 0 0 0 0 0 0 0  75E   INCRN,S=DI,IF %SIGNSAVE *GO TO FPDW.NORMALIZE;             4267.000
07B8 B 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  7BC   IF ALUZ *GO TO $+4;                                       4268.000
07B9 0 0 0 0 0 1 F 0 00 0 000 0 0 0 0 0 0 0 0        T=%T;                                                     4269.000
07BA 0 0 0 0 3 1 1 0 00 0 000 0 0 0 0 0 0 0 0        S=%DI;                                                    4270.000
07BB 0 5 0 0 3 0 0 0 00 0 05F 0 0 0 0 0 0 0 0  75E   NOD=DI,*GO TO FPDW.NORMALIZE;                              4271.000
07BC 0 0 0 4 0 5 F 0 05 0 F00 0 0 0 0 0 0 0 0        T=R(ZERO)-T,SETCAR;                                       4272.000
07BD 0 4 0 4 3 5 1 0 06 0 F0B 0 0 0 0 0 0 0 0  7BB   S=R(ZERO)-DI,USECAR,*GO TO $-2;                         4273.000
                                                                                                          4274.000
                                                MPFD.CRACK                                                4275.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    MPFD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

07BE 0 0 0 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0          S=SNIBL,TNIBL;                    4276.000

07BF 0 0 0 0 1 0 0 0 00 2 080 0 0 0 0 0 0 0 0          TNIBR;                            4277.000

07C0 0 1 0 4 0 D F 0 02 0 0F0 0 0 0 0 0 0 0 0          *JUMPJ,T=@0FFFFFFF&T;             4278.000

                                                  CLRS.DUD                           4279.000

07C1 0 1 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0          *JUMPJ,CLRS;                      4280.000
```

```
                  SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                              MPFD
PC    T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                        4281.000
                                                 ($+1)                                  4282.000
                                                 FPDW.MPY.TAB                           4283.000
07C3 0 0 0 0 0 D 0 0 02 0 080 0 0 0 0 0 0 0 0          NOD=@08000000&T;        TEST OVERFLOW CONDITION    4284.000
07C4 0 0 0 3 0 D 0 0 02 0 01D 0 0 0 0 0 0 0 0          NOD=@00000001&T,CLDNU;  TEST SHIFTED OUT BIT       4285.000
07C5 B 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  7C9     IF ALUZ *GO TO MPY.TAB;                            4286.000
07C6 B 4 0 0 1 0 0 0 01 0 088 0 0 0 0 0 0 0 0  7C8     SHIFTS(SRL),IF ALUZ *GO TO $+2;                    4287.000
07C7 0 4 0 0 1 E 8 0 0B 0 709 0 0 0 0 0 0 0 0  7C9     R(TBL1)=S,INCRN,*GO TO MPY.TAB;                    4288.000
07C8 0 4 3 0 1 E 8 0 00 0 709 0 0 0 0 0 0 0 0  7C9     R(TBL1)=S,DECRN,*GO TO MPY.TAB;                    4289.000
                                                 MPY.TAB                                4290.000
                                                 **             S AND T CONTAINS 1*     4291.000
07C9 0 0 0 0 0 3 8 0 00 0 800 0 0 0 0 0 0 0 0          R(TBL2)=S+T;         2*        4292.000
07CA 0 0 0 0 0 3 8 0 00 0 900 0 0 0 0 0 0 0 0          R(TBL3)=S+T;         3*        4293.000
07CB 0 0 0 0 0 3 8 0 00 0 A00 0 0 0 0 0 0 0 0          R(TBL4)=S+T;         4*        4294.000
07CC 0 0 0 0 0 3 8 0 00 0 B00 0 0 0 0 0 0 0 0          R(TBL5)=S+T;         5*        4295.000
07CD 0 0 0 0 0 3 8 0 00 0 C00 0 0 0 0 0 0 0 0          R(TBL6)=S+T;         6*        4296.000
07CE 0 0 0 0 0 3 8 0 00 0 D00 0 0 0 0 0 0 0 0          R(TBL7)=S+T;         7*        4297.000
07CF 0 1 0 0 0 3 8 0 00 0 F07 0 0 0 0 0 0 0 0          *JUMPJ,R(TBL8)=S+T,CLRS;      8*  4298.000
                                                 ADJ.B1                                 4299.000
07D0 2 4 0 0 1 0 0 0 01 3 D22 0 0 0 0 0 0 0 0  7D2     SHIFTD(SLLD),IF NCTRO *GO TO $+2;                  4300.000
07D1 0 4 0 4 3 3 F 0 05 D D05 0 0 0 0 0 0 0 0  7D5     T=R(TBL7)+DI,SETCAR,OTHERBANK,*GO TO ADJ.B2+2;     4301.000
07D2 0 1 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0          T=DI,*JUMPJ;                                       4302.000
                                                 ADJ.B2                                 4303.000
07D3 2 4 0 0 1 0 0 0 01 3 D22 0 0 0 0 0 0 0 0  7D2     SHIFTD(SLLD),IF NCTRO *GO TO $-1;                  4304.000
07D4 0 0 0 4 3 3 F 0 05 D F00 0 0 0 0 0 0 0 0          T=R(TBL8)+DI,SETCAR,OTHERBANK;                     4305.000
07D5 0 1 0 0 4 3 1 0 06 0 F00 0 0 0 0 0 0 0 0          S=S+R(ZERO),USECAR,*JUMPJ;                         4306.000
                                                 CLRS.OB.DUD                            4307.000
```

```
          S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    M P F D
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

07D6 0 1 0 0 1 0 0 0 00 D 007 0 0 0 0 0 0 0 0      *JUMPJ,OTHERBANK,CLRS;                    4308.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MPFD
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                        4309.000
**                                                                      4311.000
**          HARDWARE ASSISTED FUNCTION                                  4312.000
**                                                                      4313.000
**                T-REG CONTAINS MULTIPLIER                             4314.000
**                S/T-REGS WILL CONTAIN PRODUCT                         4315.000
**                        LEAST SIGNIFICANT HEX DIGIT IN T              4316.000
**                        WILL BE USED TO SELECT APPROPRIATE            4317.000
**                        MULTIPLE OF THE MULTIPLICAND                  4318.000
**                MPROM-FUNCTION                                        4319.000
**                                                                      4320.000
**                  PRIOR      T-REG       REGISTER     NEXT           4321.000
**                FUNCTION  BITS(28-31)  SELECTED  FUNCTION            4322.000
**                   +           0          F(0*)        +             4323.000
**                   +           1          7(1*)        +             4324.000
**                   +           2          8(2*)        +             4325.000
**                   +           3          9(3*)        +             4326.000
**                   +           4          A(4*)        +             4327.000
**                   +           5          B(5*)        +             4328.000
**                   +           6          C(6*)        +             4329.000
**                   +           7          D(7*)        +             4330.000
**                   +           8          E(8*)        +             4331.000
**                   +           9          D(7*)        -             4332.000
**                   +           A          C(6*)        -             4333.000
**                   +           B          B(5*)        -             4334.000
**                   +           C          A(4*)        -             4335.000
**                   +           D          9(3*)        -             4336.000
**                   +           E          8(2*)        -             4337.000
**                   +           F          7(1*)        -             4338.000
**                   -           0          7(1*)        +             4339.000
**                   -           1          8(2*)        +             4340.000
**                   -           2          9(3*)        +             4341.000
**                   -           3          A(4*)        +             4342.000
**                   -           4          B(5*)        +             4343.000
**                   -           5          C(6*)        +             4344.000
**                   -           6          D(7*)        +             4345.000
**                   -           7          E(8*)        +             4346.000
**                   -           8          D(7*)        -             4347.000
**                   -           9          C(6*)        -             4348.000
**                   -           A          B(5*)        -             4349.000
**                   -           B          A(4*)        -             4350.000
**                   -           C          9(3*)        -             4351.000
**                   -           D          8(2*)        -             4352.000
**                   -           E          7(1*)        -             4353.000
**                   -           F          F(0*)        -             4354.000
**                IF LAST REDUCTION CYCLE RESULTS                       4355.000
**                WITH A (-) FUNCTION, ONE MORE                         4356.000
**                REDUCTION CYCLE IS DONE ADDING                        4357.000
**                REG7(1*)                                              4358.000
**                                                                      4359.000
**                NHMPREV IS PRIOR ALU FUNCTION TEST                    4360.000
**                                                                      4361.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                          M P F D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                4362.000

                                                    FPSW.MPY                                                    4364.000

07D7 0 4 0 0 1 0 0 0 00 2 08B 0 0 0 0 0 0 0 0  7DB          TNIBR,*GO TO MP.CYCLES+3;                          4365.000

                                                    MP.CYCLES                                                   4366.000

07D8 0 0 0 0 1 0 0 0 00 2 080 0 0 0 0 0 0 0 0               TNIBR;                                             4367.000

07D9 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4368.000

07DA 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4369.000

07DB 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4370.000

07DC 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4371.000

07DD 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4372.000

07DE 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4373.000

07DF 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4374.000

07E0 6 0 2 3 4 3 1 0 00 2 08B 0 0 0 0 0 0 0 0               S=SNIBR+R(0),TNIBR,MPROM,MPY;                      4375.000

07E1 A 1 0 0 1 0 0 0 00 0 200 0 0 0 0 0 0 0 0               IF %NHMPREV *JUMPJ;                                4376.000

07E2 0 0 0 3 4 3 1 0 00 2 780 0 0 0 0 0 0 0 0               S=SNIBR+R(TBL1),TNIBR;                             4377.000

07E3 0 0 0 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0               S=SNIBL,TNIBL;                                     4378.000

07E4 0 0 0 0 1 E 0 0 00 0 000 0 0 0 0 0 0 0 0               NOD=S;              CYCLE REQUIRED FOR MPROM RESET 4379.000

07E5 6 1 2 0 4 0 0 0 00 0 000 0 0 0 0 0 0 0 0               NOD=R(0),MPROM,*JUMPJ;                             4380.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 75 MODE INSTRUCTIONS
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    4381.000
                                                    **                                                              4383.000
                                                    **      UNBLOCK EXTERNAL INTERRUPTS -UEI-, SECONDARY DECODE(N=00)  4384.000
                                                    **                                                              4385.000

                                                                                                                    4386.000

                                                    UEI                                                             4387.000

07E6 0 5 7 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0 7F0           *LINK CHECK.CPU.MODE;                                   4388.000

07E7 0 0 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0               SET(UNBLOCK); ENABLE INTERRUPTS                         4389.000

07E8 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0 7EC           *GO TO UEI.BEI.EXIT;                                    4390.000

                                                                                                                    4391.000
                                                    **                                                              4392.000
                                                    **      BLOCK EXTERNAL INTERRUPTS -BEI-, SECONDARY DECODE(N=00)  4393.000
                                                    **                                                              4394.000

                                                                                                                    4395.000

                                                    BEI                                                             4396.000

07E9 0 6 0 0 1 0 0 0 00 0 C76 0 0 0 0 0 0 0 0 C76           *GOTO BEI.TEST;                                         4397.000

07EA 0 0 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0       BEI.1   SET(UNBLOCK);  REFRESH BLOCK TIMEOUT COUNT              4398.000

07EB 0 0 0 0 1 0 0 0 00 0 106 0 0 0 0 0 0 0 0               RESET(UNBLOCK); DISABLE INTERRUPTS                      4399.000

                                                    UEI.BEI.EXIT                                                    4400.000

07EC 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0 829           *GO TO FETCH.RETURN;                                    4401.000


                                                                                                                    4402.000

                                                    ALTER.EXTENDED.STATUS                                           4403.000

07ED 0 0 0 0 0 0 F 0 02 0 FA0 0 0 0 0 0 0 0 0               T=@F8000000&T;  GET CURRENT CC'S FROM STATUS             4404.000

07EE 0 0 0 0 0 A 1 0 00 0 000 0 0 0 0 0 0 0 0               S=S:T; COMBINE NEW STATUS & CURRENT STATUS               4405.000

07EF 0 1 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0               SETXCC(S), *JUMPJ; SET STATUS                           4406.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU UNDEFINED EXITS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                4407.000
                                                CHECK.CPU.MODE                                  4408.000
07F0 A 4 0 4 2 3 7 2 00 0 307 0 0 0 0 0 0 0 0  7F7      MARIX=R(X)+IO, IF %MODE75 *GO TO UNDEF.55;  4409.000
07F1 9 1 0 0 4 0 0 4 00 0 980 0 0 0 0 0 0 0 0           NOD=R(R), IF %PRIVBIT:EXFLAG *JUMPJ;    4410.000
                                                PRIV.EXECUTE.ERROR                              4411.000
07F2 9 4 0 0 1 0 0 0 00 0 B85 0 0 0 0 0 0 0 0  7F5      IF %PRIVBIT *HOP UNDEF.75;IF PRIV BIT=1, ASSUME EXECUTE VIOL  4412.000
07F3 0 6 0 0 1 0 0 0 00 0 9AC 0 0 0 0 0 0 0 0  9AC      *GO TO PRIV.VIOL; EXIT TO ERROR IF NOT PRIVILEGED  4413.000
                                                                                                4414.000
                                                ***                                             4415.000
                                                *** HALFWORD UNDEFINED 75 MODE INSTRUCTIONS (RHF NOT ADJUSTED)  4416.000
                                                ***                                             4417.000
                                                                                                4418.000
                                                UNDEF.75.1                                      4419.000
07F4 0 0 0 0 1 0 0 0 00 5 104 0 0 0 0 0 0 0 0           TOGRHF, RESET(EXFLAG);  ADJUST RIGHTHAND FLAG  4420.000
                                                UNDEF.75                                        4421.000
07F5 0 4 0 1 1 E F 0 02 0 088 0 0 0 0 0 0 0 0  7F8      T=UNDEF.75.FLG, *HOP UNDEF.55+1; SET UNDEF 75 FLAG IN CPU STATUS  4422.000
                                                CHECK.NOT.55.ENBL                               4423.000
07F6 9 4 0 0 1 0 0 0 00 0 F31 0 0 0 0 0 0 0 0  7F1      IF %ENBL55 *HOP CHECK.CPU.MODE+1;       4424.000
                                                *       SETCPU OR RDSTS IS UNDEFINED IF 55 OPTION JUMPER IS REMOVED.  4426.000
                                                UNDEF.55                                        4428.000
07F7 0 0 0 2 1 E F 0 02 0 100 0 0 0 0 0 0 0 0           T=UNDEF.55.FLG; SET UNDEF 55 MODE FLAG IN CPU STATUS  4429.000
07F8 0 6 7 0 1 0 0 0 00 0 596 0 0 0 0 0 0 0 0  596      *LINK UPDATE.CPU.STATUS;                4430.000
07F9 0 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  7FE      *GOTO UNDEF.XX;                         4431.000
                                                *                                               4432.000
                                                CALM.INT.TEST                                   4433.000
                                                *                                               4434.000
07FA 1 4 0 0 1 0 0 0 00 0 DAC 0 0 0 0 0 0 0 0  7FC        IF FFINT *HOP $+2;          GOTO EXTG  4435.000
07FB 1 4 0 0 1 0 0 0 00 0 DFD 0 0 0 0 0 0 0 0  7FD        IF INTRENA *HOP $+2;        RETURN TO CALM PROCESSOR  4436.000
07FC 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1        *GOTO EXTG;                             4437.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU UNDEFINED EXITS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
07FD 0 6 0 0 1 0 0 0 00 0 990 0 0 0 0 0 0 0 0  990          *GOTO CALM.RET;                              4438.000
                                                *                                                        4439.000
                                                UNDEF.XX                                                 4440.000
                                                *                                                        4441.000
07FE 0 0 0 0 2 0 0 0 00 0 004 0 0 0 0 0 0 0 0                NOD=10,RESET(HIREG);                        4442.000
07FF 0 6 0 0 1 0 0 0 00 0 9AA 0 0 0 0 0 0 0 0  9AA          *GOTO UNDEF+1;                               4443.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU I/O & INTR CTL
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                                                    4444.000
                                                *                                                                   4445.000
                                                *** PRIMARY DECODE Q=3F, EI, DI, RI, DAI, AI, TD, CD ***            4446.000
                                                *** SIO, TIO, STPIO, RSCHNL, HIO, GRIO, RSCTL, ECWCS ***            4447.000
                                                *** WCWCS, ECI, DCI, ACI, & DACI ***                                4448.000
                                                *                                                                   4449.000
                                                (@800)                                                              4450.000
                                                INTR.CTL.IO                                                         4451.000
0800 0 6 0 0 1 0 0 0 0C 0 D2C 0 0 0 0 0 0 0 0 D2C       RSTRHF,*GOTO IPU.CTL;     CHECK FOR IPU MODE              4452.000
0801 0 6 0 0 1 0 0 0 0C 0 460 0 0 0 0 0 0 0 0 460     RSTRHF,*GO TO LEXTPROC;                                       4453.000
                                                *                                                                   4454.000
                                                INTR.CTL.RTN                                                        4455.000
                                                *                                                                   4456.000
0802 9 4 0 3 1 E 1 0 0C 0 B84 0 0 0 0 0 0 0 0 804       S=SMIBR, RSTRHF, IF %PRIVBIT *HOP $+2;                      4457.000
0803 0 6 0 0 1 0 0 0 00 0 9AC 0 0 0 0 0 0 0 0 9AC       *GO TO PRIV.VIOL;                                           4458.000
0804 0 2 0 1 1 E F 0 00 0 804 0 0 0 0 0 0 0 0           T=SLEFT,*JUMPD,SET(HIPEG);                                  4459.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 55/75 MODE INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR                                                    4461.000
                                                *                                                     4462.000
                                                *** CONTROL CLASS ***                                 4463.000
0805 9 2 0 0 4 0 F 6 00 5 BE0 0 0 0 0 0 0 0 0       CTL         T=R(K0), TOGRHF, IF %LATERR *JUMPD ;  4464.000
0806 0 6 0 0 1 0 0 0 00 5 460 0 0 0 0 0 0 0 0  460             TOGRHF, *GOTO LEXTPROC ;              4465.000
                                                *                                                     4466.000
                                                *** LOAD CONTROL SWITCHES ***                         4467.000
0807 0 0 0 3 1 E F 0 02 0 780 0 0 0 0 0 0 0 0       LCS         T=@00000078; SET CSWS ADDR (RIGHT SHIFTED) 4468.000
0808 A 4 0 0 1 0 0 0 00 2 A0A 0 0 0 0 0 0 0 0  80A             TNIBL, IF %RHFLAG *GO TO $+2; SET T = @00000780 4469.000
0809 0 0 0 0 1 0 0 0 0C 0 F05 0 0 0 0 0 0 0 0                  RSTRHF, SET(FLAG); REMEMBER RIGHT HAND FLAG 4470.000
080A 0 0 0 0 0 0 E 0 00 0 007 0 0 0 0 0 0 0 0                  FULLMAR=T, CLRS; LOAD CSWS ADDR IN MAR      4471.000
080B 0 0 0 0 1 E F 0 1C 1 080 0 0 0 0 0 0 0 0       LCS2        T=S, READ, FRCWORD; FETCH CSWS              4472.000
080C 9 4 0 0 0 1 1 0 00 F DEE 0 0 0 0 0 0 0 0  80E             S=%T(7E), IF %FLAG *HOP $+2; TEST FOR REMEMBERED RHFLAG 4473.000
                                                *                AND LOAD CSWS MASK INTO S (S=@FFFF0000) 4474.000
080D 0 0 0 0 1 E 3 0 00 C 705 0 0 0 0 0 0 0 0                  MAR=S, SETRHF, RESET(FLAG);                 4475.000
080E 0 5 1 0 5 D 9 0 00 0 029 0 0 0 0 0 0 0 0  829             R(R)=S&DI, SETCC(#), *GOTO FETCH.RETURN ;   4476.000
                                                *                                                     4477.000
                                                *** EXTEND SIGN ***                                   4478.000
                                                *       (TIMING: 6 OR 7 CLOCKS)                       4479.000
080F 0 0 1 0 1 9 9 0 00 0 000 0 0 0 0 0 0 0 0       ES          R(R)=%S, SETCC(#) ;                   4480.000
0810 2 5 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829             IF ALUNEG *GOTO FETCH.RETURN ;        4481.000
0811 0 5 1 0 1 E 9 0 00 0 029 0 0 0 0 0 0 0 0  829             R(R)=S, SETCC(#) *GOTO FETCH.RETURN ; 4482.000
                                                *                                                     4483.000
                                                *** ROUND REGISTER ***                                4484.000
                                                *       (TIMING: 6 OR 7 CLOCKS)                       4485.000
0812 0 0 0 3 1 E F 0 02 0 010 0 0 0 0 0 0 0 0       RND         T=1 ;                                 4486.000
0813 A 5 1 0 4 0 0 4 00 0 829 0 0 0 0 0 0 0 0  829             NOD=R(R), SETCC(#), IF %ALUNEG *GOTO FETCH.RETURN ; 4487.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    55/75 MODE INSTRUCTIONS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0814 0 5 1 4 0 3 9 4 00 0 029 0 0 0 0 0 0 0 0  829              R(R)=R(R)+T, SETCC(#), *GOTO FETCH.RETURN ;          4488.000
                                                *                                                                   4489.000
                                                *** EXCHANGE REGISTERS ***                                          4490.000
                                                *               (TIMING: 5 CLOCKS)                                  4491.000
0815 0 0 0 0 4 0 1 4 00 0 001 0 0 0 0 0 0 0 0    XCR            S=R(R), SETCC(AL) ;                                  4492.000
0816 0 4 0 0 0 0 9 0 19 0 08A 0 0 0 0 0 0 0 0  81A              R(R)=T, FETCHPC, *HOP XCRM2 ;                        4493.000
                                                *                                                                   4494.000
                                                *** EXCHANGE REGISTERS MASKED ***                                   4495.000
                                                *               (TIMING: 6 CLOCKS)                                  4496.000
0817 0 0 0 0 4 0 1 4 00 0 000 0 0 0 0 0 0 0 0    XCRM           S=R(R) ;                                             4497.000
0818 0 0 1 0 4 0 1 0 00 0 400 0 0 0 0 0 0 0 0                   S=S&R(4), SETCC(#) ;                                 4498.000
0819 0 0 0 4 0 0 9 0 19 0 480 0 0 0 0 0 0 0 0                   R(R)=R(4)&T, FETCHPC ;                               4499.000
081A A 3 4 0 1 E B 5 00 9 A40 0 0 0 0 0 0 0 0    XCRM2          R(S)=S, DECODE(4), SHIFTIO, IF %RHFLAG *JUMPZ ;      4500.000
081B 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   DECODE(#), IF NOEXTUNIV *JUMPD ;                     4501.000
081C 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1              *GOTO EXTG ;                                         4502.000
                                                **              **1-CLOCK DELAY**                                   4503.000
081D 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0    DELAY1         *JUMPJ;                                              4504.000
                                                *                                                                   4505.000
                                                *** ADD REGISTER & SUBTRACT REGISTER MASKED (ADRM & SURM) ***       4506.000
081E 0 0 0 4 0 F F 4 00 0 001 0 0 0 0 0 0 0 0    RRM1           T=R(R)#T, SETCC(AL);                                 4507.000
081F 0 4 1 4 0 D 9 0 19 0 48A 0 0 0 0 0 0 0 0  82A              R(R)=R(4)&T, SETCC(#), FETCHPC, *HOP SWAP;           4508.000
                                                *                                                                   4509.000
                                                *** REGISTER-REGISTER CLASS ***                                     4510.000
0820 9 2 0 0 4 0 F 5 00 5 BE0 0 0 0 0 0 0 0 0    REGREG         TOGRHF, T=R(S), IF %LATERR *JUMPD ;                  4511.000
0821 0 6 0 0 1 0 0 0 00 5 460 0 0 0 0 0 0 0 0  460              TOGRHF, *GOTO LEXTPROC ;                             4512.000
                                                *                                                                   4513.000
0822 0 4 1 4 0 F 9 4 19 0 08A 0 0 0 0 0 0 0 0  82A  RR1         R(R)=R(R)#T, SETCC(#), FETCHPC, *HOP SWAP ;          4514.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 55/75 MODE INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + R A D  ADDR
0823 0 4 1 0 0 5 9 0 19 0 08A 0 0 0 0 0 0 0 0  82A  TKN          R(R)=S-T, SETCC(#), FETCHPC, *HOP SWAP ;                4515.000
0824 0 0 0 4 0 F F 4 00 0 000 0 0 0 0 0 0 0 0       CMR          T=R(R)#T ;                                              4516.000
0825 0 4 1 4 0 D F 0 19 0 48A 0 0 0 0 0 0 0 0  82A               T=R(4)&T, SETCC(#), FETCHPC, *HOP SWAP ;                4517.000
0826 0 0 0 4 0 F F 4 00 0 000 0 0 0 0 0 0 0 0       RRM          T=R(R)#T ;                                              4518.000
0827 0 4 1 4 0 D 9 0 19 0 48A 0 0 0 0 0 0 0 0  82A               R(R)=R(4)&T, SETCC(#), FETCHPC, *HOP SWAP ;             4519.000
0828 0 4 1 4 0 F 0 4 19 0 08A 0 0 0 0 0 0 0 0  82A  CAR          NOD=R(R)&T, SETCC(#), FETCHPC, *HOP SWAP ;              4520.000
                                                    *                                                                4521.000
                                                    *** COMMON HALFWORD INSTRUCTION EXIT POINT ***                   4522.000
                                                    FETCH.RETURN                                                     4523.000
0829 0 0 0 0 1 0 0 0 19 0 084 0 0 0 0 0 0 0 0       NOP          FETCHPC,RESET(HIREG);                                   4524.000
082A A 3 4 0 1 0 0 0 00 9 A40 0 0 0 0 0 0 0 0       SWAP         DECODE(4), SHIFTIO, IF %RHFLAG *JUMPZ ;                 4525.000
082B 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), IF NOEXTUNIV *JUMPD ;                        4526.000
082C 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1               *BRANCH EXTG ;                                          4527.000
                                                    *                                                                4528.000
                                                    *** TRANSFER REGISTER NEGATIVE MASKED (TRNM) ***                 4529.000
082D 0 0 0 0 0 5 F 0 00 0 001 0 0 0 0 0 0 0 0       TRNM         T=S-T, SETCC(AL);                                       4530.000
082E 0 4 1 4 0 D 9 0 00 0 409 0 0 0 0 0 0 0 0  829               R(R)=R(4)&T, SETCC(#), *GO TO FETCH.RETURN;             4531.000
                                                    *                                                                4532.000
                                                    *** BIT-REGISTER ***                                             4533.000
082F 9 2 0 6 1 9 F 0 00 5 8E0 0 0 0 0 0 0 0 0       BITREGS      T=%BMG, TOGRHF, IF %LATERR *JUMPD ;                     4534.000
0830 0 6 0 0 1 0 0 0 00 5 460 0 0 0 0 0 0 0 0  460               TOGRHF, *GOTO LEXTPROC ;                                4535.000
                                                    *                                                                4536.000
0831 0 4 1 4 0 3 B 5 19 0 083 0 0 0 0 0 0 0 0  833  ABR          R(S)=R(S)+T, SETCC(#), FETCHPC, *HOP SWAP2 ;            4537.000
                                                    *                                                                4538.000
0832 0 0 0 6 4 B 0 5 19 0 481 0 0 0 0 0 0 0 0       TOR          NOD=%BMG&R(S), SETCC(BIT), FETCHPC ;                    4539.000
0833 A 3 4 0 1 0 0 0 00 9 A40 0 0 0 0 0 0 0 0       SWAP2        DECODE(4), SHIFTIO, IF %RHFLAG *JUMPZ ;                 4540.000
0834 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), IF NOEXTUNIV *JUMPD ;                        4541.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU 55/75 MODE INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0835 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1          *GOTO EXTG ;                                          4542.000

                                                 *                                                             4543.000

0836 0 0 0 6 4 B 0 5 19 0 481 0 0 0 0 0 0 0 0    SBRZBR     MOD=%RMG&R(S), SETCC(BIT), FETCHPC ;                  4544.000

0837 A 3 4 6 4 F R 5 00 9 A40 0 0 0 0 0 0 0 0               R(S)=RMG&R(S),DFCODE(4),SHIFTIO,IF %KHFLAG *JUMPZ ;    4545.000

0838 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0               DFCODE(#), IF NOEXTUNIV *JUMPD ;                      4546.000

0839 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1          *GOTO EXTG ;                                          4547.000

                                                 *                                                             4548.000

                                                 *** IMMEDIATE CLASS (Q=32) ***                                4549.000

083A 9 2 0 0 4 0 1 4 0C 8 BE0 0 0 0 0 0 0 0 0    IMMED      S=R(R), RSTRHF, PCTOMAR,IF %LATERR *JUMPD ;           4550.000

083B 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0 460          *BRANCH LEXTPROC ;                                    4551.000

083C 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0    LASI       FETCHPC ;                                             4552.000

083D 0 3 1 4 2 F 9 4 00 E 000 0 0 0 0 0 0 0 0               R(R)=R(F)#IO(SE), SETCC(#), *JUMP7 ;                  4553.000

083E 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0    CI         FETCHPC ;                                             4554.000

083F 0 3 0 4 2 5 0 4 00 F 701 0 0 0 0 0 0 0 0               MOD=R(R)-IO(SE), SETCC(C), *JUMP7 ;                   4555.000

                                                 *                                                             4556.000

                                                 *** SHIFT SINGLE (Q=1B,1C,1D) ***                             4557.000

0840 5 4 0 1 2 0 0 0 02 C 1F2 0 0 0 0 0 0 0 0 842 SHIFTS     NL=@001F0000&IO, IF %LATERRW *HOP $+2 ;               4558.000

0841 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0 460          *GOTO LEXTPROC ;                                      4559.000

0842 0 0 0 0 4 0 1 4 12 5 000 0 0 0 0 0 0 0 0              S=R(R), SAVFSIGN, TOGRHF ;                            4560.000

0843 B 2 3 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              DECRN, IF ALU7 *JUMPD ;                               4561.000

0844 A 4 2 0 1 0 0 0 01 0 904 0 0 0 0 0 0 0 0 844          SHIFTS(#), IF %NCTR7 *GOTO $ ;                        4562.000

0845 0 2 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPD ;                                              4563.000

                                                 *                                                             4564.000

                                                 *** SHIFT DOUBLE (Q=1E,1F) ***                                4565.000

0846 5 4 0 1 2 0 0 0 02 C 1F8 0 0 0 0 0 0 0 0 848 SHIFTD     NL=@001F0000&IO, IF %LATERRW *HOP $+2 ;               4566.000

0847 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0 460          *GOTO LEXTPROC ;                                      4567.000

0848 1 4 0 0 4 0 1 4 12 0 BD8 0 0 0 0 0 0 0 0 848          S=R(R), SAVFSIGN, IF BIRUSY *HOP $ ;                  4568.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          5 5 / 7 5  M O D E  I N S T R U C T I O N S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0849 2 2 3 0 4 0 6 6 00 5 900 0 0 0 0 0 0 0 0                    DI=R(R0), DECRN, TOGRHF, IF NCTRZ *JUMPD ;                4569.000
084A A 4 2 0 1 0 0 0 01 3 90A 0 0 0 0 0 0 0 0  84A               SHIFTD(#), IF %NCTR7 *GO TO $ ;                           4570.000
084B 0 2 0 0 3 0 8 6 00 0 000 0 0 0 0 0 0 0 0                    R(R0)=DI, *JUMPD ;                                        4571.000
084C 0 0 0 0 1 E 9 0 19 0 080 0 0 0 0 0 0 0 0        SHIFTX1     R(R)=S, FETCHPC ;                                         4572.000
084D A 3 4 0 1 0 0 0 00 9 A40 0 0 0 0 0 0 0 0                    DECODE(4), SHIFTI0, IF %RHFLAG *JUMPZ ;                   4573.000
084E 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), IF NOEXTUNTV *JUMPD ;                          4574.000
084F 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1               *GOTO EXTG ;                                              4575.000
0850 0 0 0 0 1 E F 0 00 0 101 0 0 0 0 0 0 0 0        SHIFTX2     T=S, SETCC(V) ;          -LEFT ARITH SHIFT-               4576.000
0851 5 4 0 4 0 D 9 0 02 1 7F3 0 0 0 0 0 0 0 0  853               R(R)=@7FFFFFFFRT, IF %STGNSAVE *HOP $+2 ;                 4577.000
0852 0 0 0 0 0 A 9 0 02 0 800 0 0 0 0 0 0 0 0                    R(R)=@80000000:T ;                                        4578.000
0853 0 5 0 0 1 0 0 0 00 0 029 0 0 0 0 0 0 0 0  829               *GOTO FETCH.RETURN ;                                      4579.000
                                                 *                                                                     4580.000
                                                 *** NORMALIZE ***                                                     4581.000
                                                 *       (TIMING: 12 + 3(N) )                                          4582.000
0854 3 6 0 0 4 0 1 4 00 C 460 0 0 0 0 0 0 0 0  460  NOR          S=R(R), IF LATERRW *GOTO LEXTPROC ;                       4583.000
0855 0 0 0 0 1 9 0 0 00 5 000 0 0 0 0 0 0 0 0                    NOD=%S, TOGRHF ;                                          4584.000
0856 8 4 0 3 1 E 8 5 02 0 00C 0 0 0 0 0 0 0 0  85C               R(S)=0, IF ALUZ *HOP NOR3 ;                               4585.000
0857 8 4 0 3 1 E 8 5 02 0 40C 0 0 0 0 0 0 0 0  85C               R(S)=@00000040, IF ALUZ *HOP NOR3 ;                       4586.000
0858 0 5 7 0 1 E 1 0 00 0 01D 0 0 0 0 0 0 0 0  81D               S=S , *LINK DELAY1;                                       4587.000
0859 5 4 0 0 1 0 0 0 00 5 00C 0 0 0 0 0 0 0 0  85C  NOR2         IF %NORC *GOTO NOR3 ;                                     4588.000
085A 0 0 0 2 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0                    S=SNIBL ;                                                 4589.000
085B 0 4 0 4 1 4 8 5 00 0 009 0 0 0 0 0 0 0 0  859               R(S)=R(S)-1, *GOTO NOR2 ;                                 4590.000
085C 0 5 0 0 1 E 9 0 00 0 029 0 0 0 0 0 0 0 0  829  NOR3         R(R)=S, *GOTO FETCH.RETURN ;                              4591.000
                                                 *                                                                     4592.000
                                                 *** NORMALIZE DOUBLE ***                                              4593.000
085D 3 6 0 0 4 0 1 4 00 C 460 0 0 0 0 0 0 0 0  460  NORD         S=R(R), IF LATERRW *GOTO LEXTPROC ;                       4594.000
085E 0 0 0 0 4 0 F 6 00 5 000 0 0 0 0 0 0 0 0                    T=R(R0), TOGRHF ;                                         4595.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL  32/75  CPU                                                    55/75  MODE  INSTRUCTIONS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

085F 8 4 0 0 1 9 0 0 00 0 007 0 0 0 0 0 0 0 0  867                 NOD=%S, IF ALUZ *GOTO NORD3 ;                          4596.000
0860 0 0 0 0 0 1 0 0 00 0 000 0 0 0 0 0 0 0 0                       NOD=%T ;                                               4597.000
0861 C 4 0 0 1 E C 0 02 0 400 0 0 0 0 0 0 0 0  86D                 NU=@40000000, IF NALUZ *HOP NORD5;                     4598.000
0862 5 4 0 0 1 9 0 0 00 0 005 0 0 0 0 0 0 0 0  865                 IF ALUZ *GOTO NORD2 ;                                  4599.000
0863 1 4 0 0 1 0 0 0 00 0 7FD 0 0 0 0 0 0 0 0  86D                 IF SONET0 *GOTO NORD5 ;                                4600.000
0864 0 4 0 0 1 0 0 0 00 0 00A 0 0 0 0 0 0 0 0  86A                 *GOTO NORD4 ;                                          4601.000
0865 0 0 0 3 1 E B 5 02 0 400 0 0 0 0 0 0 0 0       NORD2          R(S)=@00000040 ;                                       4602.000
0866 0 5 0 0 1 0 0 0 00 0 029 0 0 0 0 0 0 0 0  829                 *GOTO FETCH.RETURN ;                                   4603.000
0867 8 5 0 0 1 0 0 0 00 0 076 0 0 0 0 0 0 0 0  876 NORD3           IF ALUZ *GOTO NORD6 ;                                  4604.000
0868 0 0 0 0 1 E C 0 02 0 400 0 0 0 0 0 0 0 0                       NU=@40000000 ;                                         4605.000
0869 1 4 0 0 1 0 0 0 00 0 7FD 0 0 0 0 0 0 0 0  86D                 IF SONET0 *GOTO NORD5 ;                                4606.000
086A 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0       NORD4          S=T ;                                                  4607.000
086B 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0                       T=0 ;                                                  4608.000
086C 0 0 0 0 1 E C 0 02 0 380 0 0 0 0 0 0 0 0                       NU=@38000000 ;                                         4609.000
086D 0 4 0 0 1 E 0 0 00 0 00F 0 0 0 0 0 0 0 0  86F NORD5           NOD=S, *GOTO $+2 ;                                     4610.000
086E 0 0 3 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0                       DECRN ,S=SNIBL, TNIBL ;          -NIBBLE SHIFT LEFT-   4611.000
086F 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                       *NOP ;                                                 4612.000
0870 3 5 0 0 1 0 0 0 00 5 06E 0 0 0 0 0 0 0 0  86E                 IF NORC *GOTO NORD5+1 ;                                4613.000
0871 0 0 0 0 0 0 8 6 00 0 000 0 0 0 0 0 0 0 0                       R(RO)=T ;                -NORMALIZATION COMPLETE-      4614.000
0872 0 0 0 0 1 E 9 0 00 0 000 0 0 0 0 0 0 0 0                       R(R)=S ;                                               4615.000
0873 0 0 0 0 1 D F 0 02 0 FF7 0 0 0 0 0 0 0 0                       T=@FF000000&N, CLRS ;                                  4616.000
0874 0 0 0 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0                       S=SNIBL, TNIBL ;                                       4617.000
0875 0 0 0 2 1 E 1 0 00 2 000 0 0 0 0 0 0 0 0                       S=SNIBL, TNIBL ;                                       4618.000
0876 0 5 0 0 1 E B 5 00 0 029 0 0 0 0 0 0 0 0  829 NORD6           R(S)=S, *GOTO FETCH.RETURN ;                           4619.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U M E M O R Y R E F E R E N C E L O A D
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    4620.000
                                        * PRIMARY DECODE FOR MEMORY REFERENCE 'LOAD' TYPE INSTRUCTIONS               4622.000
                                        * -ANMW,ORMW,EOMW,CAMW,CMMW,LW,LMW,LNW,ADMW,SUMW,MPMW,DVMW,LF,ARMW-         4623.000
                                        * OP CODES = 84, 88, 8C, 90, 94, AC, B0, B4, B8, BC, C0, C4,                4624.000
                                        * OP CODES = CC, AND EP.                                                    4625.000
                                        *                                                                           4626.000

                                                                                                                    4627.000

                                        (@877)                                                                      4628.000

                                        MRLOAD                                                                      4629.000

0877 3 6 0 4 2 3 7 2 0C C 460 0 0 0 0 0 0 0 0  460      MARIX=R(X)+IO, RSTRHF, IF LATERRW *GO TO LEXTPROC;          4630.000

0878 0 0 0 0 1 0 0 0 1C 0 080 0 0 0 0 0 0 0 0           READ; FETCH OPERAND OR INDIRECT WORD                        4631.000

0879 A 2 0 0 1 0 0 0 00 0 C00 0 0 0 0 0 0 0 0           IF %INDIR *JUMPD; TEST FOR INDIRECT                         4632.000

                                        MRLOAD.INDR                                                                 4633.000

087A 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0           MARIX=R(DIX)+DI; WAIT FOR INDIRECT WORD                     4634.000

087B 0 0 0 0 1 0 0 0 1C 0 080 0 0 0 0 0 0 0 0           READ; FETCH OPERAND OR INDIRECT WORD                        4635.000

087C 9 4 0 0 1 0 0 0 00 0 55F 0 0 0 0 0 0 0 0  87E      IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2; TEST PREVIOUS READ 4636.000

087D 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR;                        4637.000

087E A 2 0 0 1 0 0 0 00 0 C00 0 0 0 0 0 0 0 0           IF %INDIR *JUMPD; EXIT IF NOT ANOTHER INDIRECT CYCLE        4638.000

087F 5 5 0 0 1 0 0 0 00 0 07A 0 0 0 0 0 0 0 0  87A      IF %EXTLW *GO TO MRLOAD.INDR; LOOP IF NO EXTERNAL LOCAL     4639.000

0880 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      *GO TO EXTPROC1; EXIT TO EXTERNAL LOCAL PROCESSING          4640.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MEMORY REFERENCE LOAD

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4641.000
                                                    * SECONDARY DECODE OF MEMORY REFERENCE LOAD, ADD, AND SUBTRACT                   4643.000
                                                    * -LW,ADMW,SUMW-                                                                4644.000
                                                    * OP CODES = AC, B8, AND BC;  0 = 2B, 2E, AND 2F                                4645.000
                                                    *                                                                               4646.000
                                                                                                                                    4647.000
                                                    MRLAS                                                                           4648.000
0881 5 4 0 0 4 0 1 4 00 D 005 0 0 0 0 0 0 0 0  885          S=R(R), IF %DWORD *GO TO MRLAS1; GET REGISTER OPERAND IN S              4649.000
                                                                                                                                    4650.000
                                                    MRLAS.DWORD                                                                     4651.000
                                                    MRLOG.DWORD                                                                     4652.000
                                                    ANM.DWORD                                                                       4653.000
                                                                                                                                    4654.000
0882 0 5 7 0 4 0 1 6 00 0 08A 0 0 0 0 0 0 0 0  88A          S=R(R0), *LINK FETCH.MSW; LOAD LEAST SIGNIFICANT REG OPRND IN S         4655.000
0883 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0               FETCHPC; FETCH ($+8)                                                    4656.000
0884 0 3 1 4 3 F 9 4 06 0 000 0 0 0 0 0 0 0 0               R(R)=R(R)#DI, SETCC(#), USECAP, *JUMPZ; COMPUTE MOST SIGNIFICANT        4657.000
                                                    MRLAS1                                                                          4658.000
0885 5 4 0 0 1 0 0 0 19 E 069 0 0 0 0 0 0 0 0  889          FETCHPC, IF %HWORD *HOP MRLAS.BYTE.WORD;                                4659.000
                                                    MRLAS.HWORD                                                                     4660.000
0886 0 3 1 0 3 F 9 0 00 E 000 0 0 0 0 0 0 0 0               R(R)=S#DI(SE), SETCC(#), *JUMPZ;                                        4661.000
                                                                                                                                    4662.000
                                                    *                                                                               4663.000
                                                    * SECONDARY DECODE FOR MEMORY REFERENCE LOGICAL INSTRUCTIONS                    4664.000
                                                    * -ORMW,EOMW-                                                                   4665.000
                                                    * OP CODES = 88 AND 8C; 0 = 22 AND 23.                                          4666.000
                                                    *                                                                               4667.000
                                                                                                                                    4668.000
                                                    MRLOG                                                                           4669.000
0887 3 4 0 0 4 0 1 4 00 D 002 0 0 0 0 0 0 0 0  882          S=R(R), IF DWORD *GO TO MRLOG.DWORD;                                    4670.000
0888 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0               FETCHPC; FETCH ($+8)                                                    4671.000
                                                    MRLOG.B.H.W                                                                     4672.000
                                                    MRLAS.BYTE.WORD                                                                 4673.000
0889 0 3 1 0 3 F 9 0 00 0 000 0 0 0 0 0 0 0 0               R(R)=S#DI, SETCC(#), *JUMPZ;                                            4674.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    M E M O R Y  R E F E R E N C E  L O A D
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4675.000
                                                *                                                                   4676.000
                                                * SECONDARY DECODE FOR MEMORY REFERENCE LOAD NEGATIVE INSTRUCTION   4677.000
                                                * -LNW-, OP CODE = B4 ; Q = 2D                                      4678.000
                                                *                                                                   4679.000

                                                                                                                    4680.000

                                                LN                                                                  4681.000

088A 5 4 0 0 1 0 0 0 00 D 005 0 0 0 0 0 0 0 0  885          IF %DWORD *GO TO MRLAS1;  S REG =0                      4682.000

                                                LN.DWORD                                                            4683.000

                                                LM.DWORD                                                            4684.000

088B 0 5 7 0 1 0 0 0 00 0 08A 0 0 0 0 0 0 0 0  88A          *LINK FETCH.MSW; INITIAL S REG =0                       4685.000

088C 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0              FETCHPC; FETCH ($+8)                                    4686.000

088D 0 3 1 0 3 F 9 0 06 0 000 0 0 0 0 0 0 0 0              R(R)=S#DI, SETCC(#), USECAR, *JUMPZ; COMPUTE MOST SIGNIFICANT 4687.000

                                                                                                                    4688.000
                                                *                                                                   4689.000
                                                * SECONDARY DECODE FOR MEMORY REFERENCE LOAD MASKED INSTRUCTION     4690.000
                                                * -LMW-, OP CODE = B0; Q = 2C                                       4691.000
                                                *                                                                   4692.000

                                                                                                                    4693.000

                                                LM                                                                  4694.000

088E 3 4 0 0 4 0 1 0 00 D 408 0 0 0 0 0 0 0 0  888          S=R(4), IF DWORD *GO TO LM.DWORD;                       4695.000

088F 0 5 0 0 1 0 0 0 00 0 085 0 0 0 0 0 0 0 0  885          *GO TO MRLAS1;                                          4696.000

                                                                                                                    4697.000
                                                *                                                                   4698.000
                                                * SECONDARY DECODE FOR 'AND' MEMORY INSTRUCTION                     4699.000
                                                * -ANMW-, OP CODE = B4; Q = 21                                      4700.000
                                                *                                                                   4701.000

                                                                                                                    4702.000

                                                ANM                                                                 4703.000

0890 3 5 0 0 4 0 1 4 00 D 082 0 0 0 0 0 0 0 0  882          S=R(R), IF DWORD *GO TO ANM.DWORD;                      4704.000

0891 5 4 0 7 4 D 1 4 00 E 004 0 0 0 0 0 0 0 0  894          S=@FFFFFF00&R(R), IF %HWORD *GO TO ANM.WORD.BYTE; SET UP FOR BYTE 705.000

                                                ANM.HWORD                                                           4706.000

0892 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0              FETCHPC;  FETCH ($+8)                                   4707.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                              M E M O R Y   R E F E R E N C E   L O A D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0893 0 3 0 4 3 D 9 4 0D 0 001 0 0 0 0 0 0 0 0          R(R,RH)=R(R)&DI, SETCC(AL), *JUMP7; PERFORM HALFWORD 'AND'        4708.000

                                                ANM.WORD.BYTE                                                            4709.000

0894 5 4 0 0 1 0 0 0 19 F 087 0 0 0 0 0 0 0 0  897      FETCHPC, IF %BYTE *HOP ANM.WORD;  FETCH ($+8)                   4710.000

                                                ANM.BYTE                                                                 4711.000

0895 0 0 0 4 3 0 F 4 00 0 001 0 0 0 0 0 0 0 0          T=R(R)&DI, SETCC(AL); PERFORM BYTE 'AND'                         4712.000

0896 0 3 0 0 0 A 9 0 00 0 000 0 0 0 0 0 0 0 0          R(R)=S:T, *JUMPZ; COMBINE NEW BYTE 3 WITH INITIAL BYTES 0-2.     4713.000

                                                ANM.WORD                                                                 4714.000

0897 0 3 0 4 3 D 9 4 00 0 001 0 0 0 0 0 0 0 0          R(R)=R(R)&DI, SETCC(AL), *JUMPZ;  PERFORM WORD 'AND'.            4715.000

                                                                                                                         4716.000
                                                *                                                                        4717.000
                                                * SECONDARY DECODE FOR COMPARE ARITHMETIC MEMORY INSTRUCTION            4718.000
                                                * -CAMW-, OP CODE = 90, Q = 24                                          4719.000
                                                *                                                                        4720.000

                                                                                                                         4721.000

                                                CAM                                                                      4722.000

0898 5 4 0 0 1 0 0 0 00 D 00C 0 0 0 0 0 0 0 0  89C      IF %DWORD *GO TO CAM.B.H.W;  S-REG INITIALLY =0.                 4723.000

                                                CAM.DWORD                                                                4724.000

0899 0 5 7 0 1 0 0 0 00 0 08F 0 0 0 0 0 0 0 0  89F      *LINK CAM.FETCH.MSW; GO FETCH MOST SIGNIFICANT WORD              4725.000

089A 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0          FETCHPC;  FETCH ($+8)                                            4726.000

089B 0 3 0 4 3 5 0 4 06 0 701 0 0 0 0 0 0 0 0          NOD=R(R)-DI, USECAR, SETCC(C), *JUMPZ; COMPUTE MOST SIGNIFICANT  4727.000

                                                CAM.B.H.W                                                                4728.000

089C 5 4 0 0 1 0 0 0 19 E 08F 0 0 0 0 0 0 0 0  89E      FETCHPC, IF %HWORD *HOP CAM.WORD.BYTE;  FETCH ($+8)              4729.000

                                                CAM.HWORD                                                                4730.000

089D 0 3 0 4 3 5 0 4 00 E 701 0 0 0 0 0 0 0 0          NOD=R(R)-DI(SE), SETCC(C), *JUMPZ;                               4731.000

                                                CAM.WORD.BYTE                                                            4732.000

089E 0 3 0 4 3 5 0 4 00 0 701 0 0 0 0 0 0 0 0          NOD=R(R)-DI, SETCC(C), *JUMPZ;                                   4733.000

                                                                                                                         4734.000
                                                *                                                                        4735.000
                                                * SECONDARY DECODE FOR COMPARE MASKED MEMORY INSTRUCTION                4736.000
                                                * -CMMW-, OP CODE = 94; Q = 25                                          4737.000
                                                *                                                                        4738.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    MEMORY REFERENCE LOAD
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                4739.000

                                                CMM                                             4740.000

089F 5 4 0 0 4 0 1 0 00 D 404 0 0 0 0 0 0 0 0  8A4      S=R(4), IF %DWORD *GO TO CMM.R.H.W; LOAD MASK IN S-REG  4741.000

                                                CMM.DWORD                                       4742.000

08A0 0 5 7 0 1 0 0 0 00 0 0C4 0 0 0 0 0 0 0 0  8C4      *LINK CMM.FETCH.MSW; GO GET MOST SIGNIFICANT WORD       4743.000

08A1 0 0 0 0 0 D 0 0 00 0 301 0 0 0 0 0 0 0 0           NOD=S&T, SETCC(D); MASK LEAST SIGNIFICANT WORD  4744.000

                                                CMM.WORD.BYTE                                   4745.000

08A2 0 0 0 4 3 8 F 4 19 0 080 0 0 0 0 0 0 0 0           T=R(R)!DI, FETCHPC; COMPUTE MOST SIGNIFICANT & FETCH (S+8)  4746.000

                                                CMM.EXIT                                        4747.000

08A3 0 3 0 0 0 D 0 0 00 0 201 0 0 0 0 0 0 0 0           NOD=S&T, SETCC(E), *JUMPZ; MASK MOST SIGNIFICANT  4748.000

                                                CMM.R.H.W                                       4749.000

08A4 5 4 0 0 1 0 0 0 00 E 002 0 0 0 0 0 0 0 0  8A2      IF %HWORD *GO TO CMM.WORD.BYTE;                 4750.000

                                                CMM.HWORD                                       4751.000

08A5 0 4 0 4 3 8 F 4 19 E 083 0 0 0 0 0 0 0 0  8A3      T=R(R)!DI(SE), FETCHPC, *HOP CMM.EXIT;  FETCH (S+8)  4752.000

                                                                                                4753.000
                                                *                                               4754.000
                                                *** SHIFT COUNT ZEROES ***                      4755.000
                                                *   PRIMARY DECODE OF OP CODE = 68, Q = 1A    *  4756.000
                                                *       (TIMING: 10+2(N-1) )                    4757.000

                                                                                                4758.000

                                                SC7                                             4759.000

08A6 3 6 0 0 4 0 1 4 00 C 460 0 0 0 0 0 0 0 0  460      S=R(R), IF LATERRW *GOTO LEXTPROC ;              4760.000

08A7 0 0 0 0 1 E 1 0 00 5 201 0 0 0 0 0 0 0 0           S=S, SETCC(E), TOGRHF ;                         4761.000

08A8 B 4 0 3 1 E F 0 02 0 008 0 0 0 0 0 0 0 0  8AB      T=0, IF ALUZ *HOP SC72 ;                        4762.000

08A9 2 4 0 1 1 E 1 0 00 0 808 0 0 0 0 0 0 0 0  8AB      S=SLEFT, IF ALUNEG *GOTO SC72 ;                 4763.000

08AA 0 4 0 3 0 3 F 0 02 0 019 0 0 0 0 0 0 0 0  8A9      T=1+T, *HOP $-1 ;                               4764.000

                                                SC72                                            4765.000

08AB 0 0 0 0 0 0 E 0 00 0 00D 0 0 0 0 0 0 0 0           FULLMAR=T, CLDNU ;                              4766.000

08AC 0 0 0 0 1 E 9 0 00 0 000 0 0 0 0 0 0 0 0           R(R)=S ;                                        4767.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MEMORY REFERENCE LOAD
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

08AD 0 5 0 0 1 0 B 5 00 0 029 0 0 0 0 0 0 0 0 829 R(S)=MAR, *GOTO FETCH.RETURN ; 4768.000

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
MEMORY REFERENCE LOAD

SEL 32/75 CPU

PC	T S M A B + D R Y X PCH + B D M + B A D ADDR		
			4769.000
			4771.000
		*	4772.000
		* LOAD FILE, -LF-, SECONDARY DECODE OF OP CODE = CC, Q=33	4773.000
		*	4774.000
			4775.000
		LF	
		FR(TEMP3)=@FFFFFFE0&MAR; GET FILE BOUNDARY ADDRESS	4776.000
08AE	6 0 0 7 1 D B 5 00 0 E00 0 0 0 0 0 0 0 0		
		S=@FC000000:%I0; GET ONES-COMPLEMENT OF REGISTER ADDR IN 6-8	4777.000
08AF	0 0 0 0 2 7 1 0 02 0 FC0 0 0 0 0 0 0 0 0		
		NU=SLEFT; ONES-COMPLEMNT OF REG ADDR PLUS 8 TO N BITS 4-7	4778.000
08B0	0 0 0 1 1 E C 0 00 0 000 0 0 0 0 0 0 0 0		
		S=@00000004, *HOP LF1; SET MEMORY ADDR INCREMENT VALUE	4779.000
08B1	0 4 0 3 1 E 1 0 02 0 044 0 0 0 0 0 0 0 0	8B4	
			4780.000
		LOOP.LF	
		MAR=R(TEMP3):T; COMPUTE NEXT MEMORY ADDRESS	4781.000
08B2	0 0 0 4 0 A 3 0 00 0 D00 0 0 0 0 0 0 0 0		
		R(NCTR)=DI, READ, FRCWORD, DECRN; LOAD A REGISTER & FETCH NEXT	4782.000
08B3	0 0 3 0 3 0 8 3 1C 1 080 0 0 0 0 0 0 0 0		
			4783.000
		LF1	
		NOD=@07000000&N; TEST FOR REG ADDR =R(7)	4784.000
08B4	0 0 0 0 1 D 0 0 02 0 070 0 0 0 0 0 0 0 0		
		T=S+MAR, IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2; BUMP MEM ADDR	4785.000
08B5	9 4 0 0 1 3 F 0 00 0 557 0 0 0 0 0 0 0 0	8B7	4787.000
	*	AND TEST FOR PREVIOUS MEMORY ERRORS.	
		NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS	4789.000
08B6	0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0	530	
		T=@0000001F&T, IF NALUZ *HOP LOOP.LF; GET BIT 27-31 OF NEXT	4790.000
08B7	C 4 0 3 0 D F 0 02 0 1F2 0 0 0 0 0 0 0 0	8B2	4792.000
	*	MEMORY ADDRESS AND BRANCH IF REG ADDR NOT =R(7).	
			4794.000
		LF.EXIT	
		FETCHPC; FETCH ($+8)	4795.000
08B8	0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0		
		R(7)=DI, *JUMPZ;	4796.000
08B9	0 3 0 0 3 0 8 0 00 0 700 0 0 0 0 0 0 0 0		

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MEMORY REFERENCE MSW FETCH

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    4797.000
                                                                                                                    4799.000
                                *                                                                                   4800.000
                                *              DOUBLE WORD MEMORY REFERENCE INSTRUCTION, FETCH                       4801.000
                                *                 MOST SIGNIFICANT WORD                                             4802.000
                                *                                                                                   4803.000
                                *       ENTRY PARAMETERS :                                                          4804.000
                                *         A FETCH OF THE LEAST SIGNIFICANT WORD IS IN PROGRESS.                      4805.000
                                *         MAR = ADDRESS OF THE LEAST SIGNIFICANT WORD (DWORD ADDR).                  4806.000
                                *         S = LEAST SIGNIFICANT REGISTER OPERAND.                                    4807.000
                                *                                                                                   4808.000
                                *       EXIT PARAMETER :                                                            4809.000
                                *         T = LEAST SIGNIFICANT WORD.                                               4810.000
                                *                                                                                   4811.000

                                FETCH.MSW                                                                           4812.000

08BA 0 0 0 0 1 0 3 0 1C 1 080 0 0 0 0 0 0 0 0       MAR=MAR,READ,FRCWORD; INTER LOCK &FETCH MOST SIGNIFICANT WORD  4813.000

08BB 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0       T=DI;  TRANSFER LEAST SIGNIFICANT WORD TO T.                   4814.000

08BC 0 0 0 0 0 F 6 6 05 0 301 0 0 0 0 0 0 0 0       R(R0)=S#T, SETCAR, SETCC(D); COMPUTE LEAST SIGNIFICANT WORD    4816.000

08BD 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0       IF %OPNOKESP:OPTIMEOUT:OPRNOPE *JUMPJ;                         4817.000

08BE 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0 530   NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR     4818.000

                                                                                                                    4819.000

                                ARM.FETCH.MSW                                                                       4820.000

                                CAM.FETCH.MSW                                                                       4821.000

08BF 0 0 0 0 1 0 3 0 1C 1 080 0 0 0 0 0 0 0 0       MAR=MAR,READ,FRCWORD; INTER LOCK & FETCH MOST SIGNIFICANT WORD 4822.000

08C0 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0       T=DI;  TRANSFER LEAST SIGNIFICANT WORD TO T                    4823.000

08C1 0 0 0 4 0 F F 6 05 0 301 0 0 0 0 0 0 0 0       T=R(R0)#T, SETCAR, SETCC(D); COMPUTE LEAST SIGNIFICANT WORD    4825.000

08C2 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0       IF %OPNOKESP:OPTIMEOUT:OPRNOPE *JUMPJ;                         4826.000

08C3 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0 530   NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR     4827.000

                                                                                                                    4828.000

                                CMM.FETCH.MSW                                                                       4829.000

08C4 0 0 0 0 1 0 3 0 1C 1 080 0 0 0 0 0 0 0 0       MAR=MAR,READ,FRCWORD; INTER LOCK & FETCH MOST SIGNIFICANT WORD 4830.000

08C5 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0       T=DI; TRANSFER LEAST SIGNIFICANT WORD TO T                     4831.000

08C6 0 0 0 4 0 8 F 6 00 0 000 0 0 0 0 0 0 0 0       T=R(R0)!T; COMPUTE LEAST SIGNIFICANT WORD                      4833.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
                                                                  M E M O R Y   R E F E R E N C E   M S W   F E T C H
S E L   3 2 / 7 5   C P U
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

08C7 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0          IF %OPNORESP:OPTIMEOUT:OPRNDPE *JUMPJ;                                  4834.000

08C8 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530     NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR                4835.000

                                                                                                                        4836.000
                                                                                                                        4837.000
                                                   *                                                                    4838.000
                                                   * ZERO MEMORY, -ZM-                                                  4839.000
                                                   * SECONDARY DECODE OF OP CODE = F8, Q=3E                             4840.000
                                                   *
                                                                                                                        4841.000

                                                   ZM                                                                   4842.000

08C9 0 0 0 0 1 E F 0 1E 0 080 0 0 0 0 0 0 0 0          T=S, WRITE; S REG. =0 INITIALLY                                  4843.000

08CA A 5 0 0 1 0 0 0 00 0 CDD 0 0 0 0 0 0 0 0  8DD     IF %INDIR *GO TO ST; GO TO COMMON PROCESSING                     4844.000

                                                   ZM.INDIRECT                                                          4845.000

08CB 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0          MARTX=R(DIX)+DI; WAIT FOR INDIRECT WORD                          4846.000

08CC 0 0 0 0 1 E F 0 1E 0 080 0 0 0 0 0 0 0 0          T=S, WRITE; WRITE ZEROS OR FETCH INDIRECT WORD                   4847.000

08CD 9 4 0 0 1 0 0 0 00 0 55F 0 0 0 0 0 0 0 0  8CF     IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2; TEST PREVIOUS READ      4848.000

08CE 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530     NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR         4849.000

08CF A 4 0 0 1 0 0 0 00 0 CDD 0 0 0 0 0 0 0 0  8DD     IF %INDIR *GO TO ST; EXIT IF NOT ANOTHER INDIRECT                4850.000

08D0 5 5 0 0 1 0 0 0 00 0 0CB 0 0 0 0 0 0 0 0  8CB     IF %EXTLW *GO TO ZM.INDIRECT;                                    4851.000

08D1 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D     *GO TO EXTPROC1; EXIT TO EXTERNAL LOCAL PROCESSING               4852.000
```

```
            S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    M E M O R Y  R E F E R E N C E  S T O R E
PC  T S M A B + D B  Y X PCH + B D M + B A D  ADDR

                                                                                                                        4853.000
                                                *                                                                       4855.000
                                                * PRIMARY DECODE FOR MEMORY REFERENCE, STORE MASKED INSTRUCTION         4856.000
                                                * -STMW-, OP CODE = D8; Q = 36.                                         4857.000
                                                *                                                                       4858.000

                                                                                                                        4859.000

                                                STM                                                                     4860.000

08D2 0 4 0 0 4 0 1 0 00 0 403 0 0 0 0 0 0 0 0  8D3      S=R(4), *GO TO MRSTORE; LOAD MASK IN TO S REG                   4861.000

                                                                                                                        4862.000
                                                *                                                                       4863.000
                                                * PRIMARY DECODE FOR MEMORY REFERENCE STORE AND STORE FILE INSTRUCTIONS 4864.000
                                                * -STW,STF-, OP CODE = D4 AND DC; Q = 35 AND 37                         4865.000
                                                *                                                                       4866.000

                                                                                                                        4867.000

                                                MRSTORE .                                                               4868.000

08D3 3 6 0 4 2 3 7 2 0C C 460 0 0 0 0 0 0 0 0  460      MARIX=R(X)+I0, RSTRHF, IF LATERPW *GO TO LEXTPROC;              4869.000

08D4 0 0 0 0 4 F F 7 1E 0 080 0 0 0 0 0 0 0 0           T=S#R(RD), WRITE;  STORE OPERAND OR FETCH INDIRECT WORD.        4870.000

08D5 A 2 0 0 0 0 C 0 00 F C00 0 0 0 0 0 0 0 0           NU=T(ZE), IF %INDIR *JUMPD; TEST FOR INDIRECT AND CLEAR N REG.  4871.000

                                                MRSTORE.INDR                                                            4872.000

08D6 3 6 7 4 3 3 7 1 00 0 989 0 0 0 0 0 0 0 0  989      MARIX=R(DIX)+DI, IF DWORD *LINK STORE.DOUBLE.INDIR;             4873.000
                                                *          WAIT FOR INDIRECT WORD                                       4875.000

08D7 0 0 0 0 4 F F 7 1F 0 080 0 0 0 0 0 0 0 0           T=S#R(RD), WRITE;  STORE OPERAND OR FETCH INDIRECT WORD.        4877.000

                                                STORE.DOUBLE.INDIR.RTN                                                  4878.000

08D8 9 4 0 0 1 0 0 0 00 0 55A 0 0 0 0 0 0 0 0  8DA      IF %OPNORESP:OPTIMEOUT:OPRNDPF *HOP $+2; TEST PREVIOUS READ    4879.000

08D9 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR;                           4880.000

08DA A 2 0 0 0 1 C 0 00 F C00 0 0 0 0 0 0 0 0           NU=%T(ZE), IF %INDIR *JUMPD; EXIT IF NOT ANOTHER INDIRECT CYCLE 4881.000

08DB 5 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  8D6      IF %EXTLW *GO TO MRSTORE.INDR; LOOP IF NO EXTERNAL LOCAL       4882.000

08DC 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      *GO TO EXTPROC1; EXIT TO EXTERNAL LOCAL PROCESSING             4883.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MEMORY REFERENCE STORE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                4884.000
                                                *                                                               4886.000
                                                * SECONDARY DECODE FOR STORE AND STORE MASKED INSTRUCTIONS      4887.000
                                                * -STW,STMW-, OP CODES = D4 AND D8; Q = 35 AND 36               4888.000
                                                *                                                               4889.000

                                                                                                                4890.000

                                                ST                                                              4891.000

08DD 5 5 0 0 1 0 0 0 00 D 0E1 0 0 0 0 0 0 0 0  8F1      IF %DWORD *GO TO STORE.EXIT;                           4892.000

                                                ST.DWORD                                                        4893.000

                                                STM.DWORD                                                       4894.000

08DE 0 0 0 0 4 F F 4 1E 1 080 0 0 0 0 0 0 0 0            T=S#R(R), WRITE, FRCWORD; STORE MOST SIGNIFICANT WORD  4895.000

08DF 9 4 0 0 1 0 0 0 00 0 551 0 0 0 0 0 0 0 0  8F1      IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2; TEST PREVIOUS ERRORS  4896.000

08E0 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS  4897.000

                                                STORE.EXIT                                                      4898.000

08F1 0 3 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0            FETCHPC, *JUMP7;  FETCH ($;8)                          4899.000

                                                                                                                4900.000
                                                *                                                               4901.000
                                                * STORE FILE, -STF=, SECONDARY DECODE OF OP CODE =DC, Q=37      4902.000
                                                *                                                               4903.000

                                                STF                                                             4905.000

08E2 0 0 0 0 2 7 1 0 02 0 FC0 0 0 0 0 0 0 0 0            S=@FC000000:%I0; GET ONES COMLEMENT OF REG ADDRESS IN BITS 6-8  4906.000

08E3 1 4 0 1 1 E C 0 00 0 7EC 0 0 0 0 0 0 0 0  8FC      NU=SLEFT, IF PROTV *HOP STF.EXIT;                      4907.000

08E4 6 0 0 7 1 D 6 5 00 0 F00 0 0 0 0 0 0 0 0            FR(TEMP3)=@FFFFFFE0&MAR; GET FILE BOUNDARY ADDRESS     4908.000

08E5 0 4 0 3 1 E 1 0 02 0 048 0 0 0 0 0 0 0 0  8E8      S=@00000004, *HOP STF1; SET MEMORY ADDR BUMP VALUE      4909.000

                                                LOOP.STF                                                        4910.000

08E6 0 0 0 4 3 A 3 0 00 0 D00 0 0 0 0 0 0 0 0            MAR=R(TEMP3):D1; COMPUTE NEXT MEMORY ADDRESS           4911.000

08F7 0 0 0 0 4 0 F 3 1E 1 080 0 0 0 0 0 0 0 0            T=R(NCTR), WRITE, FRCWORD; STORE A REGISTER             4912.000

                                                STF1                                                            4913.000

08E8 0 0 0 0 1 D 0 0 02 0 07E 0 0 0 0 0 0 0 0            NOD=@07000000&N, CDFCRN; TEST FOR REG ADDR =R(7) AND SET  4914.000
                                                *        NEXT REGISTER ADDRESS IN N REG BITS 4-7 (ONES-COMPLEMENT)  4916.000

08E9 9 4 0 0 1 3 6 0 00 0 55B 0 0 0 0 0 0 0 0  8EB      DI=S+MAR, IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2; BUMP MEM ADDR  4918.000
                                                *        AND TEST FOR PREVIOUS MEMORY ERRORS.                   4920.000
```

```
                    S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEI  32/75  CPU                                                                  MEMORY  REFERENCE  STORE
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

08EA 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530          NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS          4922.000

08EB C 4 0 3 3 D 6 0 02 0 1F6 0 0 0 0 0 0 0 0  8E6          DI=@0000001F&DI, IF NALUZ *HOP LOOP.STF; GET BITS 27-31 OF NEW      4923.000
                                                *          MEMORY ADDRESS AND BRANCH IF REG ADDR NOT =R(7).                    4925.000

                                                STF.EXIT                                                                       4927.000

08EC 0 3 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0             FETCHPC, *JUMPZ; FETCH ($+8) AND EXIT                                4926.000

                                                                                                                               4929.000
                                                *                                                                              4930.000
                                                * SECONDARY DECODE FOR ADD REGISTER TO MEMORY INSTRUCTION                      4931.000
                                                * -ARMW-, OP CODE = E8; Q = 3A.                                                4932.000
                                                *                                                                              4933.000

                                                                                                                               4934.000

                                                ARM                                                                            4935.000

08ED 5 5 0 0 1 0 0 0 00 D 0F7 0 0 0 0 0 0 0 0  8F7          IF %DWORD *GO TO ARM.W.H.B;                                          4936.000

                                                ARM.DWORD                                                                      4937.000

08EE 0 5 7 0 1 0 0 0 00 0 08F 0 0 0 0 0 0 0 0  88F          *LINK ARM.FETCH.MSW; GO GET MOST SIGNIFICANT WORD                    4938.000

08EF 0 0 0 3 1 A 3 0 02 0 020 0 0 0 0 0 0 0 0             MAR=@00000002:MAR;  SET DWORD BIT IN MAR                             4939.000

08F0 0 0 0 0 1 0 0 0 1E 0 082 0 0 0 0 0 0 0 0             WRITE, FORCEZ; WRITE LEAST SIGNIFICANT WORD                          4940.000

08F1 0 0 0 4 3 3 1 4 06 0 001 0 0 0 0 0 0 0 0             S=R(R)+DI, USECAR, SETCC(AL); COMPUTE MOST SIGNIFICANT               4941.000

08F2 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *NOP;  REQUIRED FOR BUS TIMING                                       4942.000

08F3 9 4 0 0 1 0 0 0 00 0 555 0 0 0 0 0 0 0 0  8F5          IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2; TEST PREVIOUS ERRORS        4943.000

08F4 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530          NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR            4944.000

08F5 0 0 0 0 1 E F 0 1F 1 080 0 0 0 0 0 0 0 0             T=S, WRITE, FRCWORD; STORE LEAST SIGNIFICANT WORD                    4945.000

08F6 0 5 0 0 1 0 0 0 00 0 0E1 0 0 0 0 0 0 0 0  8F1          *GO TO STORE.EXIT;                                                   4946.000

                                                ARM.W.H.B                                                                      4947.000

08F7 3 4 0 0 1 0 0 0 00 E 009 0 0 0 0 0 0 0 0  8F9          IF HWORD *GO TO $+2;                                                 4948.000

                                                ARM.WORD.BYTE                                                                  4949.000

08F8 0 4 1 4 3 3 F 4 1F 0 08A 0 0 0 0 0 0 0 0  8FA          T=R(R)+DI, SETCC(#), WRITE, *HOP $+2; COMPUTE AND STORE RESULT       4950.000

                                                ARM.HWORD                                                                      4951.000

08F9 0 0 1 4 3 3 F 4 1E E 080 0 0 0 0 0 0 0 0             T=R(R)+DI(SE), SETCC(#), WRITE; COMPUTE AND STORE HALFWORD           4952.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU MEMORY REFFRENCE STORE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

08FA 0 5 0 0 1 0 0 0 00 0 0E1 0 0 0 0 0 0 0 0  8E1        *GO TO STORE.EXIT;
```

4953.000

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                          B I T   I N   M E M O R Y   I N S T R U C T I O N S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                        4954.000
                                                *                                                                       4956.000
                                                * SET, ZERO, AND ADD BIT IN MEMORY INSTRUCTION, -SBM,ZBM,ABM-           4957.000
                                                * PRIMARY DECODE OF OP CODES = 98, 9C, AND A0 ; Q = 26, 27, AND 28      4958.000
                                                *                                                                       4959.000

                                                                                                                        4960.000
                                                BIT.MEM                                                                 4961.000
08FB 3 6 0 4 2 3 7 2 0C C 460 0 0 0 0 0 0 0 0  460      MARIX=R(X)+IO, RSTRHF, IF LATERRW *GO TO LEXTPROC;              4962.000
08FC 0 0 0 0 1 0 0 0 1C 1 0A0 0 0 0 0 0 0 0 0           RDLOCK,FRCWORD;  FETCH OPERAND OR INDIRECT WORD                 4963.000
08FD A 2 0 6 1 9 1 0 04 0 C00 0 0 0 0 0 0 0 0           S=%BMGMR, IF %INDIR *JUMPD; LOAD BIT MASK IN S                  4964.000
                                                BIT.MEM.INDIRECT                                                        4965.000
08FE 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0           MARIX=R(DIX)+DI;                                                4966.000
08FF 0 0 0 0 1 0 0 0 1C 1 0A0 0 0 0 0 0 0 0 0           RDLOCK,FRCWORD;  FETCH OPERAND OR ANOTHER INDIRECT WORD         4967.000
0900 9 4 0 0 1 0 0 0 00 0 552 0 0 0 0 0 0 0 0  902      IF %OPNOKESP:OPTIMEOUT:OPRNDPF *HOP $+2; TEST FOR PREVIOUS ERRORS 968.000
0901 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS      4969.000
0902 A 2 0 6 1 9 1 0 04 0 C00 0 0 0 0 0 0 0 0           S=%BMGMR, IF %INDIR *JUMPD;                                     4970.000
0903 5 6 0 0 1 0 0 0 00 0 8FE 0 0 0 0 0 0 0 0  8FE      IF %EXTLW *GO TO BIT.MEM.INDIRECT; ALLOW EXTERNAL EVENT LOCAL   4971.000
0904 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      *GO TO EXTPROC1;                                                4972.000
                                                SBM                                                                     4973.000
                                                ZBM                                                                     4974.000
0905 0 0 0 0 3 F F 0 1F 1 080 0 0 0 0 0 0 0 0           T=S#DI, WRITE, FRCWORD; COMPUTE OPERAND AND STORE IN MEMORY     4975.000
                                                BIT.MEM.EXIT                                                            4976.000
0906 0 0 0 6 3 B 0 0 04 0 401 0 0 0 0 0 0 0 0           NOD=%BMGMR&DI, SETCC(BIT); PERFORM IMPLIED TEST FUNCTION        4977.000
                                                ABM.EXIT                                                                4978.000
0907 0 3 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0           FETCHPC, *JUMPZ; FETCH ($+8) AND EXIT                           4979.000
                                                ABM                                                                     4980.000
0908 0 0 0 0 3 3 F 0 1E 1 080 0 0 0 0 0 0 0 0           T=S+DI, WRITE, FRCWORD; COMPUTE OPERAND AND STORE               4981.000
0909 0 4 1 0 3 3 0 0 00 0 007 0 0 0 0 0 0 0 0  907      NOD=S+DI, SETCC(#), *GO TO ABM.EXIT; RECOMPUTE FOR SET CC'S     4982.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  79APR27
S E L  32/75  C P U                                                    B I T   I N   M E M O R Y   I N S T R U C T I O N S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                4983.000
                                                *                                                                               4985.000
                                                * TEST BIT IN MEMORY INSTRUCTION, -TBM-                                         4986.000
                                                * PRIMARY DECODE OF OP CODE = A4, Q = 29                                        4987.000
                                                *                                                                               4988.000

                                                                                                                                4989.000

                                                BIT.TBM                                                                         4990.000

090A 3 6 0 4 2 3 7 2 0C C 460 0 0 0 0 0 0 0 0  460              MARIX=R(X)+I0, RSTRHF, IF LATFRRW *GO TO LEXTPROC;              4991.000

090B 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0                   READ, FRCWORD;  FETCH OPERAND OR INDIRECT WORD                  4992.000

090C A 5 0 6 1 9 1 0 04 0 C14 0 0 0 0 0 0 0 0  914              S=%BMGMR, IF %INDIR *GO TO TBM; LOAD BIT MASK IN S              4993.000

                                                BIT.TBM.INDIRECT                                                                4994.000

090D 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0                   MARIX=R(DIX)+D1;                                                4995.000

090E 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0                   READ, FRCWORD; FETCH OPERAND OR ANOTHER INDIRECT WORD           4996.000

090F 9 4 0 0 1 0 0 0 00 0 551 0 0 0 0 0 0 0 0  911              IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2; TEST FOR PREVIOUS ERRORS 997.000

0910 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530              NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS      4998.000

0911 A 4 0 6 1 9 1 0 04 0 C04 0 0 0 0 0 0 0 0  914              S=%BMGMR, IF %INDIR *GO TO TBM;                                 4999.000

0912 5 5 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  90D              IF %EXITLW *GO TO BIT.TBM.INDIRECT; ALLOW EXTERNAL LOCAL EVENTS 5000.000

0913 0 6 0 0 1 0 0 0 00 0 450 0 0 0 0 0 0 0 0  450              *GO TO EXTPROC1; EXIT TO EXTERNAL EVENT LOCAL PROCESSING        5001.000

                                                TBM                                                                             5002.000

0914 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0                   FETCHPC; FETCH ($+8)                                            5003.000

0915 0 3 0 0 3 0 0 0 00 0 401 0 0 0 0 0 0 0 0                   NOD=SRDI, SETCC(BIT), *JUMPZ; TEST OPERAND AND EXIT             5004.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                    I N S T R U C T I O N   E X E C U T I O N
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5005.000
                                                *                                                                   5007.000
                                                * ZERO MEMORY, BRANCH & LINK, MEMORY PROTECT, BRI, LPSD, LPSDCM & JWCS 5008.000
                                                * PRIMARY DECODE OF OP CODE = F8, Q=3E                               5009.000
                                                *                                                                   5010.000

                                                                                                                    5011.000

                                                Q3E                                                                 5012.000

0916 9 2 0 4 2 3 7 2 0C 0 BE0 0 0 0 0 0 0 0 0           MARIX=R(X)+I0, RSTRHF, IF %LATERR *JUMPD ;                  5013.000

0917 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0  460      *GOTO LEXTPROC ;                                            5014.000

                                                                                                                    5015.000
                                                *                                                                   5016.000
                                                *** TRANSFER REG TO PROTECT REG ***                                  5017.000
                                                *   SECONDARY DECODE OF OP CODE = F8, Q=3E    *                     5018.000
                                                *      (TIMING: 9 CLOCKS)                                           5019.000

                                                                                                                    5020.000

                                                TRP                                                                 5021.000

0918 0 0 0 1 2 7 D 0 02 0 080 0 0 0 0 0 0 0 0           NL=@00080000:%10 ;                                          5022.000

0919 9 4 0 0 2 0 1 0 00 0 B8B 0 0 0 0 0 0 0 0  918      S=I0, IF %PRIVBIT *HOP $+2 ;                                5023.000

091A 0 5 0 0 1 0 0 0 00 0 0AC 0 0 0 0 0 0 0 0  9AC      *GOTO PRIV.VIOL ;                                           5024.000

091B 0 0 0 0 4 0 6 3 00 0 D00 0 0 0 0 0 0 0 0           R(TEMP3)=R(NCTR) ;                                          5025.000

091C 0 0 0 3 1 E 3 0 00 0 000 0 0 0 0 0 0 0 0           MAR=SNIBR;                                                  5026.000

091D 0 0 0 0 4 0 0 0 07 0 D00 0 0 0 0 0 0 0 0           NOU=R(TEMP3), WRPMAP ;                                      5027.000

091E 0 3 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0           FETCHPC, *JUMPZ ;                                           5028.000

                                                                                                                    5029.000
                                                *                                                                   5030.000
                                                *** TRANSFER PROTECT REG TO REG ***                                  5031.000
                                                *   SECONDARY DECODE OF OP CODE = F8, Q=3E    *                     5032.000
                                                *      (TIMING: 7 CLOCKS)                                           5033.000

                                                                                                                    5034.000

                                                TPR                                                                 5035.000

091F 0 0 0 1 2 7 D 0 02 0 080 0 0 0 0 0 0 0 0           NL=@00080000:%I0 ;                                          5036.000

0920 0 0 0 0 2 0 1 0 00 0 000 0 0 0 0 0 0 0 0           S=I0;                                                       5037.000

0921 0 0 0 3 1 E 3 0 00 0 000 0 0 0 0 0 0 0 0           MAR=SNI8R ;                                                 5038.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU — INSTRUCTION EXECUTION

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0922 0 0 0 5 1 9 F 0 19 0 080 0 0 0 0 0 0 0 0          T=%STATUS, FETCHPC ;                                          5039.000

0923 0 3 0 0 0 0 B 3 00 F 000 0 0 0 0 0 0 0 0          R(NCTR)=T(ZE), *JUMPZ ;                                       5040.000

                                                                                                                 5041.000
                                                                                                                 5042.000
                                                *                                                                5043.000
                                                *** SET OR CLEAR EXTENDED ADDRESSING (SEA, CEA) (Q=00)  ***      5044.000
                                                * SECONDARY DECODE OF OP CODE = 00                               5045.000
                                                *                                                                5046.000

                                                SEA                                                              5047.000

0924 0 4 0 0 1 E 1 0 02 0 046 0 0 0 0 0 0 0 0  926      S=@04000000, *HOP CHANGE.EXTENDED.MODE; SET EXT BIT          5048.000

                                                CEA                                                              5049.000

0925 0 0 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0          CLRS; CLEAR EXTENDED BIT                                      5050.000

                                                CHANGE.EXTENDED.MODE                                             5051.000

0926 0 6 7 5 1 E F 0 00 0 7ED 0 0 0 0 0 0 0 0  7ED      T=STATUS, *LINK ALTER.EXTENDED.STATUS;                       5052.000

0927 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829      *GO TO FETCH.RETURN;  EXIT CEA AND SEA INSTRUCTIONS           5053.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU BRANCH INSTRUCTIONS

```
PC   T S M A B + D R  Y Y PCH + B D M + B A D  ADDR

                                                                                                                    5054.000
                                                *                                                                   5056.000
                                                *** BRANCH FUNCTION TRUE, BRANCH CONDITION FALSE, -BFT,BCF- ***     5057.000
                                                *   PRIMARY DECODE OF OPCODE = F0, Q = 3C    *                      5058.000
                                                *                                                                   5059.000

                                                                                                                    5060.000

0928 A 2 0 4 2 3 7 2 0C 0 500 0 0 0 0 0 0 0 0       BCF             MARIX=R(X)+I0, RSTRHF, IF %CCTEST *JUMP0 ;          5061.000
                                                *       NOTE:   1. IF THIS IS A BCF INSTRUCTION AND THE CONDITION   5063.000
                                                *                  CODES ARE AS SPECIFIED, GO TO BR1.               5064.000
                                                *               2. IF THIS IS A BFT INSTRUCTION, GO TO BFT.         5065.000

                                                                                                                    5066.000

                                                BRANCH.NOT.TAKEN                                                    5067.000

0929 A 4 0 0 1 0 0 0 19 0 68B 0 0 0 0 0 0 0 0  92B                  FETCHPC, IF %LATERR7 *HOP $+2 ;  -BRANCH NOT TAKEN- 5068.000

092A 0 6 0 0 1 0 2 0 00 0 460 0 0 0 0 0 0 0 0  460                  PC=MAR,*GOTO LEXTPROC ;                             5069.000

092B 0 3 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                       *JUMP7 ;                                            5070.000

                                                *                                                                   5071.000

                                                                                                                    5072.000
                                                *** TRANSFER REG TO PSW, -TRSW- ***                                  5073.000
                                                *   PRIMARY DECODE OF OP CODE = 28, Q = 0A    *                     5074.000
                                                *      (TIMING: 8 CLOCKS)                                           5075.000

                                                                                                                    5076.000

092C A 4 0 0 4 0 3 4 0C 6 60F 0 0 0 0 0 0 0 0  92F  TRSW            MAR=R(R), RSTRHF, SDEST, IF %LATERR7 *GOTO $+3 ;    5077.000

092D A 5 0 0 1 0 0 0 00 0 A33 0 0 0 0 0 0 0 0  933                  IF %RHFLAG *GO TO BR1.ERR;                          5078.000

092E 0 6 0 0 1 0 0 0 00 C 460 0 0 0 0 0 0 0 0  460                  SETRHF, *GOTO LEXTPROC ;                            5079.000

092F 0 0 0 0 1 0 0 0 10 1 581 0 0 0 0 0 0 0 0                       FETCHV, FRCWORD, SETCC(S) ;                         5080.000

0930 0 4 0 0 1 0 0 0 00 7 006 0 0 0 0 0 0 0 0  936                  I1TOI0, *GOTO BR3 ;                                 5081.000

                                                                                                                    5082.000
                                                *                                                                   5083.000
                                                *** BRANCH UNCONDITIONAL, BRANCH CONDITION TRUE, -BU,BCT- ***       5084.000
                                                *   PRIMARY DECODE OF OP CODE = EC, Q = 38    *                     5085.000
                                                *                                                                   5086.000

                                                                                                                    5087.000

                                                BCT                                                                 5088.000

0931 A 5 0 4 2 3 7 2 0C 0 529 0 0 0 0 0 0 0 0  929                  MARIX=R(X)+I0,RSTRHF, IF %CCTEST *GOTO BRANCH.NOT.TAKEN; 5089.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU BRANCH INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0932 A 4 0 0 1 0 0 0 1D 1 684 0 0 0 0 0 0 0 0  934  BR1          FETCHV, FRCWORD, IF %LATERR7 *HOP $+2 ; -TAKEN-          5090.000

0933 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0  460  BR1.ERR      *GO TO LEXTPROC; GO SCHEDULE LATE ERROR PROCESSING       5091.000

0934 A 4 0 0 1 0 0 0 00 7 C06 0 0 0 0 0 0 0 0  936               I1TOI0, IF %INDIR *GOTO BR3 ;                           5092.000

0935 0 5 7 0 1 0 0 0 00 0 081 0 0 0 0 0 0 0 0  981  BR2          *LINK BRANCH.INDIRECT;                                  5093.000

0936 0 0 0 3 1 3 2 0 02 0 040 0 0 0 0 0 0 0 0       BR3          PC=4+MAR ;                                              5094.000

0937 2 4 0 0 1 0 0 0 19 C E89 0 0 0 0 0 0 0 0  939               FETCHPC, SETRHF, IF FC1V *HOP $+2 ;                     5095.000

0938 0 3 0 0 1 0 0 0 0C 0 000 0 0 0 0 0 0 0 0                    RSTRHF, *JUMP7 ;                                        5096.000

0939 F 3 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0                    I1TOI0, IF FALSE *JUMPZ ;                               5097.000

093A 0 0 4 0 1 0 0 0 00 9 144 0 0 0 0 0 0 0 0                    SHIFTI0, DECODE(4) ,RESET(EXFLAG);                      5098.000

093B 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), IF NOEXTUNIV *JUMPD ;                        5099.000

093C 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1               *GOTO EXTG ;                                            5100.000

                                                                                                                         5101.000
                                                 *                                                                       5102.000
                                                 *** BRANCH AND LINK, -BL- ***                                           5103.000
                                                 *   SECONDARY DECODE OF OP CODE = F8, 0 = 3E    *                       5104.000
                                                 *      (TIMING: 8 OR 9 CLOCKS)                                          5105.000
                                                 *                                                                       5106.000
                                                 *      ENTRY PARAMETER :                                                5107.000
                                                 *        T = CURRENT PC +4 (1W)                                         5108.000
                                                 *                                                                       5109.000

                                                                                                                         5110.000

093D 0 0 0 5 4 D 1 0 1D 1 98D 0 0 0 0 0 0 0 0       BL           S=STATUS&R(STMASK), FETCHV, FRCWORD, CLDNU;             5111.000

093E E 5 0 4 0 D F 0 00 7 C41 0 0 0 0 0 0 0 0  941               T=FR(PCMASK)&T, I1TOI0, IF %INDIR *GO TO BL3;           5112.000

093F 0 5 7 0 1 0 0 0 00 0 081 0 0 0 0 0 0 0 0  981  BL2          *LINK BRANCH.INDIRECT;                                  5113.000

0940 0 0 0 4 1 D 1 0 00 0 900 0 0 0 0 0 0 0 0                    S=R(STMASK)&N ; MOVE PRIOR STATUS TO S REG              5114.000

0941 4 4 7 3 1 3 2 0 02 A 043 0 0 0 0 0 0 0 0  943  BL3          PC=4+MAR, IF ENBL.AEXP *LINK BL4;                       5115.000

0942 0 5 0 0 0 A 8 0 00 8 037 0 0 0 0 0 0 0 0  937               R(0)=S:T, PCTOMAR, *GOTO BR3+1 ;                        5116.000

0943 0 1 0 0 0 A F 0 02 0 010 0 0 0 0 0 0 0 0       BL4          T=@01000000:T, *JUMPJ; SET ENABLE AEXP BIT              5117.000

                                                                                                                         5118.000
                                                 *                                                                       5119.000
                                                 *** BRANCH INCREMENT REGISTER, -BIP,BIH,BIW,BID- ***                    5120.000
                                                 *   PRIMARY DECODE OF OP CODE = F4, 0 = 3D    *                         5121.000
                                                 *                                                                       5122.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                          B R A N C H   I N S T R U C T I O N S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                          5123.000
0944 3 5 0 4 2 3 7 0 0C C 033 0 0 0 0 0 0 0 0  933  BINC   MARIX=R(0)+I0, RSTRHF, IF LATERRW *GO TO BR1.ERR;                 5124.000
0945 0 2 0 0 4 0 1 4 1D 1 080 0 0 0 0 0 0 0 0                S=R(R), FETCHV, FRCWORD, *JUMPD ;                           5125.000
0946 0 4 0 4 1 6 9 4 00 0 00A 0 0 0 0 0 0 0 0  94A  BIB          R(R)=R(R)+1, *GOTO $+4 ;                                    5126.000
0947 0 4 0 3 4 3 9 4 02 0 02A 0 0 0 0 0 0 0 0  94A  BIH          R(R)=2+R(R), *HOP $+3 ;                                     5127.000
0948 0 4 0 3 4 3 9 4 02 0 04A 0 0 0 0 0 0 0 0  94A  BIW          R(R)=4+R(R), *HOP $+2 ;                                     5128.000
0949 0 0 0 3 4 3 9 4 02 0 080 0 0 0 0 0 0 0 0       BID          R(R)=8+R(R) ;                                               5129.000
094A A 5 0 0 1 0 0 0 00 0 C50 0 0 0 0 0 0 0 0  950             IF %INDIR *GOTO BINC3 ;                                     5130.000
094B 0 6 7 0 3 0 0 0 00 0 81D 0 0 0 0 0 0 0 0  81D  BINC2      NOD=DI,*LINK DELAY1;                                        5131.000
094C 5 4 0 4 3 3 7 1 00 0 00E 0 0 0 0 0 0 0 0  94E             MARIX=R(DIX)+DI, IF %EXTLW *GOTO $+2 ;                      5132.000
094D 0 6 0 0 1 E 9 0 00 0 45D 0 0 0 0 0 0 0 0  45D             R(R)=S, *GOTO EXTPROC1 ;                                    5133.000
094E 0 0 0 0 0 0 0 0 1D 1 080 0 0 0 0 0 0 0 0                NOD=1, FETCHV, FRCWORD ;                                    5134.000
094F 2 5 0 0 4 0 0 4 00 0 C4B 0 0 0 0 0 0 0 0  94B             NOD=R(R), IF INDIR *GOTO BINC2 ;                            5135.000
0950 B 4 0 3 1 3 3 0 02 7 048 0 0 0 0 0 0 0 0  958  BINC3      MAR=4+MAR, I1TOI0, IF ALU7 *HOP BINC4 ;                     5136.000
0951 2 4 0 0 1 0 0 0 1D 1 E83 0 0 0 0 0 0 0 0  953             FETCHV, FRCWORD, IF FC1V *HOP $+2 ;                         5137.000
0952 0 3 0 3 1 3 2 0 02 0 040 0 0 0 0 0 0 0 0                PC=4+MAR, *JUMP7 ;                                          5138.000
0953 0 0 0 0 1 0 0 0 00 C 000 0 0 0 0 0 0 0 0                SETRHF ;                                                    5139.000
0954 F 3 0 3 1 3 2 0 02 7 040 0 0 0 0 0 0 0 0                PC=4+MAR, I1TOI0, IF FALSE *JUMPZ ;                         5140.000
0955 0 0 4 0 1 0 0 0 00 9 144 0 0 0 0 0 0 0 0                SHIFTI0, DECODE(4) ,RESET(EXFLAG);                          5141.000
0956 7 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                DECODE(#), IF NOEXTUNIV *JUMPD ;                            5142.000
0957 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1           *GOTO EXTG ;                                                5143.000
0958 0 0 0 0 2 0 1 0 00 0 000 0 0 0 0 0 0 0 0       BINC4      S=I0 ;                                                      5144.000
0959 0 0 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0                I1TOI0 ;                                                    5145.000
095A 0 0 0 0 1 E 5 0 19 0 080 0 0 0 0 0 0 0 0                I1=S, FETCHPC ;                                             5146.000
0958 0 3 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                *JUMPZ;                                                     5147.000

                                                                                                                          5148.000
                *                                                                                                         5149.000
                *** BRANCH FUNCTION TRUE, -BFT- ***                                                                        5150.000
                *   SECONDARY DECODE OF OP CODE = F0, Q = 3C   *                                                          5151.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
BRANCH INSTRUCTIONS

SEL 32/75 CPU
PC T S M A B + D R Y X PCH + B D M + R A D ADDR

```
                                        *       (TIMING: LOTS OF IT)                                                    5152.000
                                                                                                                        5153.000
095C 3 6 0 5 1 E 1 0 0C C 460 0 0 0 0 0 0 0 0  460  BFT         S=STATUS, RSTRHF, IF LATERRW *GOTO LEXTPROC ;           5154.000
095D 0 0 0 3 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0                   S=SNIBR ;                                               5155.000
095E 0 0 0 1 1 E F 0 00 0 000 0 0 0 0 0 0 0 0                   T=SLEFT ;                                               5156.000
095F 0 0 0 0 0 D C 0 02 0 0F0 0 0 0 0 0 0 0 0                   NU=@0F0000008T ;                                        5157.000
0960 0 0 3 0 5 0 1 0 00 0 400 0 0 0 0 0 0 0 0                   S=R(4,HWS), DECRN ;                                     5158.000
0961 B 4 3 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0  964              DECRN, IF ALU7 *GOTO BFT2 ;                             5159.000
0962 A 4 3 1 1 E 1 0 00 0 902 0 0 0 0 0 0 0 0  962              S=SLEFT, DECRN, IF %NCTRZ *GOTO $ ;                     5160.000
0963 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   *NOP ;                                                  5161.000
0964 3 5 0 4 2 3 7 2 00 2 032 0 0 0 0 0 0 0 0  932  BFT2        MARIX=R(X)+I0, IF ALUNEGW *GOTO BR1 ;                   5162.000
0965 0 3 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0       FETCH       FETCHPC, *JUMPZ ;                                       5163.000
                                                                                                                        5164.000
                                        *                                                                               5165.000
                                        *** EXECUTE REGISTER, -EXR,EXRR- ***                                            5166.000
                                        *   SECONDARY DECODE OF OP CODE = C8, Q = 32    *                               5167.000
                                        *                                                                               5168.000
                                                                                                                        5169.000
0966 9 4 0 0 2 0 3 0 00 C D88 0 0 0 0 0 0 0 0  968  EXR         MAR=I0, SETRHF, IF %EXFLAG *HOP $+2 ;                   5170.000
0967 0 5 0 0 1 0 0 0 00 0 0A6 0 0 0 0 0 0 0 0  9A6              *GOTO UNDEF1 ;                                          5171.000
0968 0 0 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0                   I1TOI0 ;                                                5172.000
0969 0 0 0 0 1 E 5 0 00 0 904 0 0 0 0 0 0 0 0                   I1=S, SET(EXFLAG) ;                                     5173.000
096A 2 4 0 0 2 0 1 0 00 7 F0D 0 0 0 0 0 0 0 0  960              S=I0, I1TOI0, IF FC1V *GOTO EXRR ;                      5174.000
096B F 3 4 0 1 E 5 0 00 0 000 0 0 0 0 0 0 0 0                   I1=S, DECODE(0), IF FALSE *JUMPZ ;                      5175.000
096C 0 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   DECODE(#), *JUMPD ;                                     5176.000
096D F 3 4 0 1 E 5 0 00 9 040 0 0 0 0 0 0 0 0       EXRR        I1=S, SHIFTI0, DECODE(4), IF FALSE *JUMPZ ;             5177.000
096E 0 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   DECODE(#), *JUMPD ;                                     5178.000
                                                                                                                        5179.000
                                        *                                                                               5180.000
                                        *** EXECUTE MEMORY, -EXM,EXMR- ***                                              5181.000
                                        *   PRIMARY DECODE OF OP CODE = A8, Q = 2A    *                                 5182.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          B R A N C H   I N S T R U C T I O N S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                  *                                                                          5183.000
                                                                                                                             5184.000
096F 3 5 0 0 4 0 C 0 00 C F33 0 0 0 0 0 0 0 0  933  FXM          MI=R(TRACE), IF LATERRW *GOTO BR1.ERR ;                     5185.000
0970 9 4 0 4 2 3 7 2 00 0 D82 0 0 0 0 0 0 0 0  972               MARIX=R(X)+I0, IF %EXFLAG *HOP EXM2 ;                      5186.000
0971 0 5 0 0 1 0 0 0 00 0 0A6 0 0 0 0 0 0 0 0  9A6               *GO TO UNDEF1 ;                                            5187.000
0972 A 0 0 0 1 0 0 0 1D 1 983 0 0 0 0 0 0 0 0        EXM2        FETCHV, FRCWORD, IF %NCTR7 IGNSTOP ; FETCH INSTRUCTION     5188.000
0973 A 4 0 0 1 0 0 0 00 0 C0A 0 0 0 0 0 0 0 0  97A               IF %INDIR *GOTO EXM3 ;                                     5189.000
0974 3 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D       IF EXTLW *GO TO EXTPROC1;  ALLOW EXTERNAL LOCAL INTERRUPTS         5190.000
0975 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0                    MARIX=R(DIX)+DI;                                           5191.000
0976 A 0 0 0 1 0 0 0 1D 1 983 0 0 0 0 0 0 0 0                    FETCHV, FRCWORD, IF %NCTR7 IGNSTOP;                        5192.000
0977 9 4 0 0 1 0 0 0 00 0 553 0 0 0 0 0 0 0 0  973               IF %UPNORESP:UPTIMEOUT:UPRNDPE *HOP EXM2+1;                5193.000
0978 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530               MI=T(7E), *GO TO CURRENT.INST.ERROR;                       5194.000
0979 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D               *GO TO EXTPROC1;                                           5195.000
097A 0 0 0 0 0 0 2 0 00 7 904 0 0 0 0 0 0 0 0        EXM3        PC=T, I1TO10, SET(EXFLAG); FLUSH PIPELINE                  5196.000
097B 2 4 0 0 1 0 0 0 19 C E8E 0 0 0 0 0 0 0 0  97E               FETCHPC, SETRHF, IF FC1V *HOP EXMR ; REFILL PIPELINE       5197.000
097C F 3 4 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0                    DECODE(0), I1TOT0, IF FALSE *JUMP7; ADVANCE PIPELINE       5198.000
097D 0 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), *JUMPD ; GO TO PRIMARY DECODE                   5199.000
097E F 3 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0        EXMR        I1TOI0, IF FALSE *JUMPZ; ADVANCE PIPELINE                  5200.000
097F 0 0 4 0 1 0 0 0 00 9 040 0 0 0 0 0 0 0 0                    SHIFTI0, DECODE(4) ; SET RIGHT HALF DECODE                 5201.000
0980 0 2 6 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                    DECODE(#), *JUMPD ; GO TO PRIMARY DECODE                   5202.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU BRANCH INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5203.000

                                                        BRANCH.INDIRECT                                             5204.000

0981 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0                   MARIX=R(DIX)+DI;                                    5205.000

0982 0 0 0 0 3 0 1 0 10 1 080 0 0 0 0 0 0 0 0                   S=DI, FETCHV, FRCWORD; SAVE INDIRECT WORD AND FETCH  5206.000
                                                        *       INSTRUCTION OR ANOTHER INDIRECT WORD.                5208.000

0983 9 4 0 0 1 0 0 0 00 0 555 0 0 0 0 0 0 0 0  985              IF %OPNOKESP:OPTIMEOUT:OPRNDPF *HOP $+2; TEST FOR PREVIOUS ERRORS 210.000

0984 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530              NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS 5211.000

0985 A 4 0 0 1 0 0 0 00 0 C08 0 0 0 0 0 0 0 0  988              IF %INDIR *GO TO BRANCH.INDIRECT.EXIT;               5212.000
                                                        *         THE INDIRECT WORD AND RETURN.                      5214.000

0986 5 4 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0  981              IF %EXTLW *GO TO BRANCH.INDIRECT; ALLOW EXTERNAL LOCAL INTERRUPT 5216.000

0987 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D              *GO TO EXTPROC1;                                    5217.000

                                                        BRANCH.INDIRECT.EXIT                                        5218.000

0988 0 1 0 0 1 E 0 0 00 0 501 0 0 0 0 0 0 0 0                   NOD=S, SETCC(S), *JUMPJ; SET CC'S AND EXIT          5219.000

                                                                                                                    5220.000

                                                        STORE.DOUBLE.INDIR                                          5221.000

0989 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0                   MARIX=R(DIX)+DI; WAIT FOR F & C BITS                5222.000

098A 5 1 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0                   MARIX=R(DIX)+DI, IF %DWORD *JUMPJ;EXIT IF C-BITS CHANGED 5223.000

098B 0 0 0 0 4 F F 6 1E 0 080 0 0 0 0 0 0 0 0                   T=S*R(RO), WRITE; COMPUTE LEAST SIGNIFICANT OPERAND  5224.000

098C 0 6 0 0 1 0 0 0 00 0 8D8 0 0 0 0 0 0 0 0  8D8              *GO TO STORE.DOUBLE.INDIR.RTN;                      5225.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU CALM INTERRUPT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5226.000
                                                *                                                                   5227.000
                                                *** CALM INSTRUCTION, PRIMARY DECODE (Q=0C) ***                     5228.000
                                                *                                                                   5229.000
                                                (@98E)                                                              5230.000
                                                CALM                                                                5231.000
                                                *                                                                   5232.000
098E 6 5 0 0 6 0 B 3 00 0 633 0 0 0 0 0 0 0 0  933      FR(TEMP1)=INTLVL,IF LATERRZ *GOTO BR1.ERR; GET CURR.POLL     5233.000
098F 0 6 0 0 1 0 0 0 00 0 7FA 0 0 0 0 0 0 0 0  7FA      *GOTO CALM.INT.TEST;              DO CALM INTERRUPT TEST    5234.000
                                                *                                                                   5235.000
                                                CALM.RET                                                            5236.000
                                                *                                                                   5237.000
0990 4 6 0 0 1 0 0 0 00 2 F83 0 0 0 0 0 0 0 0  F83      IF IPU *GOTO IPU.CALM.PROC;       IPU CALM                  5238.000
0991 0 0 0 0 1 0 0 0 00 0 804 0 0 0 0 0 0 0 0           SET(HIREG) ;                                                5239.000
0992 6 0 0 1 1 E B 2 02 0 270 0 0 0 0 0 0 0 0           FR(INTR)=@00270000 ; MASK POLLING INT LEVEL                 5240.000
0993 0 6 7 0 1 0 0 0 00 0 467 0 0 0 0 0 0 0 0  467      *LINK BUILD.INTR.ADDR;                                      5241.000
0994 6 0 0 1 4 D 1 3 02 D 7F0 0 0 0 0 0 0 0 0           S=@007F0000&FR(TEMP1), OTHERBANK ;                          5242.000
0995 A 4 0 6 1 9 F 0 02 5 AF7 0 0 0 0 0 0 0 0  997      T=%@FFFFAFFF,TOGRHF,IF %RHFLAG *HOP $+2; SET CALM,INT RTN FLAG 5243.000
0996 0 0 0 2 1 E F 0 02 0 700 0 0 0 0 0 0 0 0           T=@00007000;  SET CALM, CALM RHFLAG, & INT RTN FLAG         5244.000
0997 0 0 0 0 4 5 0 0 00 0 A00 0 0 0 0 0 0 0 0           NOD=S-R(INTR);  TEST CURRENT POLLING INT LEVEL > CALM?      5245.000
0998 0 0 0 0 4 0 1 0 00 0 B0D 0 0 0 0 0 0 0 0           S=R(INTRTAB), CLDNU;  FETCH CALM INT LEVEL FLAGS IN N       5246.000
0999 6 5 0 4 0 A B 4 00 0 BA1 0 0 0 0 0 0 0 0  9A1      FR(OFFSET)=FR(OFFSET):T, IF ALUNEG *GO TO QUEUE.CALM;BR IF CALM 5247.000
                                                TEST.TRAP.CALM                                                      5248.000
099A 0 0 0 0 1 D 0 0 02 0 040 0 0 0 0 0 0 0 0           NOD=@04000000&MAR; INT LEVEL ACTIVE FLAG?                   5249.000
099B 0 0 0 0 1 D 0 0 02 0 010 0 0 0 0 0 0 0 0           NOD=@01000000&MAR; INT LEVEL ENABLED FLAG?                  5250.000
099C C 5 0 0 4 0 0 0 00 0 CA1 0 0 0 0 0 0 0 0  9A1      NOD=R(OFFSET), IF  NALUZ *GO TO QUEUE.CALM; EXIT IF LEVEL ACTIVE 5251.000
099D B 5 0 0 4 0 0 0 00 0 CA1 0 0 0 0 0 0 0 0  9A1      NOD=R(OFFSET), IF ALUZ *GOTO QUEUE.CALM; EXIT IF NOT ENABLED 5252.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
CALM INTERRUPT

```
SEL 32/75 CPU
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

099E F 3 0 0 1 0 0 0 19 7 080 0 0 0 0 0 0 0 0          I1TOI0, FETCHPC, IF FALSE *JUMPZ; ADVANCE PIPELINE              5253.000
099F 0 6 7 0 1 0 0 0 00 0 117 0 0 0 0 0 0 0 0  117     *LINK BACKDATF.PC;                                              5254.000
09A0 0 6 0 0 0 1 C 0 00 F 572 0 0 0 0 0 0 0 0  572     NU=%T(ZE), *GO TO CALM.TRAP; FORCE ALU NEG & SET N REG BIT 00   5255.000
                                                    QUFUE.CALM                                                      5256.000
09A1 3 4 0 0 0 0 0 0 0C A 004 0 0 0 0 0 0 0 0  9A4     NOU=T,RSTRHF,IF BMUX1B *GO TO QUEUE.EXIT;EXIT IF NOT CALM RHFLAG 5257.000
09A2 1 4 0 0 1 0 0 0 00 0 D84 0 0 0 0 0 0 0 0  9A4     IF FXFLAG *HOP QUFUE.FXIT;                                      5258.000
09A3 6 0 0 0 4 A B 3 02 0 200 0 0 0 0 0 0 0 0          FR(INTRTAB)=@20000000:FR(INTRTAB); SET CALM RHFLAG              5259.000
                                                    QUFUE.FXIT                                                      5260.000
09A4 0 6 0 0 1 0 0 0 00 0 44C 0 0 0 0 0 0 0 0  44C     *GO TO REQ.INTR;                                                5261.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U U N D E F P R I V E X I T S

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5262.000
                                                    *** UNDEFINED INSTRUCTION DECODES ***                           5263.000
                                                    *                                                               5264.000
                                                    (@9A6)                                                          5265.000
                                                    *                                                               5266.000
                                                    UNDEF1                                                          5267.000
09A6 0 0 0 0 2 0 0 0 0C 0 104 0 0 0 0 0 0 0 0           NOD=I0, RSTRHF, RESET(EXFLAG) ;                             5268.000
09A7 0 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  9A9      *GO TO UNDEF ;                                              5269.000
                                                    UNDEF2                                                          5270.000
09A8 0 0 0 0 2 0 0 0 00 5 104 0 0 0 0 0 0 0 0           NOD=I0, TOGRHF, RESET(EXFLAG) ;                             5271.000
                                                    UNDEF                                                           5272.000
09A9 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5      *GOTO UNDFF.75;                                             5273.000
09AA 3 6 0 0 1 0 0 0 00 C 460 0 0 0 0 0 0 0 0  460      IF LATERRW *GO TO LFXTPROC;                                 5274.000
09AB 6 4 0 0 4 A F 7 02 0 04D 0 0 0 0 0 0 0 0  9AD      T=@04000000:FR(TRACE),*HOP SET.OTHERS;  SET UNDF INSTR FLAG 5275.000
                                                    PRTV.VTOL                                                       5276.000
09AC 6 4 0 0 4 A F 7 02 0 08D 0 0 0 0 0 0 0 0  9AD      T=@08000000:FR(TRACE),*HOP SET.OTHERS;                      5277.000
                                                    SET.OTHERS                                                      5278.000
09AD 0 0 0 0 2 0 0 0 19 0 B84 0 0 0 0 0 0 0 0           NOD=I0,FETCHPC,SET(TRACFFF);                                5279.000
09AE E 3 0 3 0 A B 7 02 0 010 0 0 0 0 0 0 0 0           FR(TRACE)=@00000001:T,IF %TRUE *JUMPZ; ADVANCE PIPELINE     5280.000
09AF 0 6 0 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0    3      *GOTO EXTPROC;                                              5281.000
```

```
          S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                          7 5  M O D E  L P S D  I N S T R U C T I O N
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5282.000

                                                                                                                    5283.000
                                                        **                                                          5284.000
                                                        **      LOAD PROGRAM STATUS DOUBLEWORD  -LPSD,LPSDCM-       5285.000
                                                        **         SECONDARY DECODE FOR Q=3E & OP CODE = F8         5286.000
                                                        **                                                          5287.000

                                                                                                                    5288.000

                                                        *                                                           5289.000

                                                        (@980)                                                      5290.000

                                                        *                                                           5291.000

                                                        LPSD                                                        5292.000

                                                        LPSDCM                                                      5293.000

0980 0 6 7 0 1 0 0 0 00 0 7F0 0 0 0 0 0 0 0 0  7F0      *LINK CHECK.CPU.MODE; GO CHECK 75 MODE, PRIV BIT, %EXFLG     5294.000

                                                        LPSD.FETCH                                                  5295.000

0981 0 0 6 0 1 0 0 0 1C 1 785 0 0 0 0 0 0 0 0           READ, FRCWORD, RESET(FLAG), DECODE(#); FETCH PSW1           5296.000

0982 A 4 0 0 1 0 0 0 00 0 C08 0 0 0 0 0 0 0 0  988      IF %INDIR *GO TO LPSD1;                                     5297.000

                                                        LPSD.INDIRECT                                               5298.000

0983 0 0 0 4 3 3 / 1 00 0 000 0 0 0 0 0 0 0 0           MARIx=R(DIX)+DI;                                            5299.000

0984 0 4 0 0 1 0 0 0 1C 1 082 0 0 0 0 0 0 0 0  982      READ, FRCWORD, *HOP LPSD.FETCH+1; FETCH INDIRECT            5300.000

0985 1 4 0 0 1 0 0 0 00 0 55D 0 0 0 0 0 0 0 0  980      IF OPNORESP:OPTIMEOUT:OPRNOPE *HOP LPSD.ERR1;               5301.000

0986 3 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      IF EXTLW *GO TO EXTPROC1; ALLOW EXTERNAL LOCAL EVENTS       5302.000

0987 A 4 0 0 1 0 0 0 00 0 C03 0 0 0 0 0 0 0 0  983      IF %INDIR *GO TO LPSD.INDIRECT; LOOP IF INDIRECT            5303.000

                                                        LPSD1                                                       5304.000

0988 6 0 0 3 4 A 1 0 02 0 030 0 0 0 0 0 0 0 0           S=@00000003:FR(PCMASK);  SET S = @0007FFFF                  5305.000

0989 0 0 0 3 1 3 3 0 02 0 040 0 0 0 0 0 0 0 0           MAR=@00000004+MAR; VECTOR TO PSW WORD 2                     5306.000

098A 0 0 0 0 3 0 8 0 1C 1 280 0 0 0 0 0 0 0 0           R(PSW1)=DI, READ, FRCWORD; FETCH PSW WORD 2                 5307.000

098B 0 0 0 0 0 0 D 2 0 00 D 204 0 0 0 0 0 0 0 0         PC=S&T, OTHERBANK, RESET(PRIV); LOAD PC FROM PSW1 & GO PRIV 5308.000

098C 9 4 0 0 0 0 1 0 00 0 55E 0 0 0 0 0 0 0 0  98E      S=T, IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2; TEST MEM ERRORS 5309.000

                                                        LPSD.ERR1                                                   5310.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    75 MODE LPSD INSTRUCTION
PC   T S M A B + D K  Y X PCH + B D M + B A D  ADDR

09BD 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530        NU=T(ZE), *GO TO CURRENT.INST.ERROR;                          5311.000
09BE 0 0 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0              SETXCC(S); SET NEW CC'S AND EXT BTT STATUS FROM PSW1          5312.000
09BF 5 0 0 0 1 0 0 0 00 2 A04 0 0 0 0 0 0 0 0              IF %ALUNEGW SET(PRIV); GO UNPRIV IF NO PRIV BIT IN PSW1       5313.000
09C0 0 0 0 0 0 D 0 0 02 0 010 0 0 0 0 0 0 0 0              NOD=@01000000&T; CHECK N.PSW1 FOR ENABLE ARITHMETIC EXCP      5314.000
09C1 0 0 0 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0              RESET(ENBL.AEXP); DISABLE ARITHMETIC EXCEPTION                5315.000
09C2 C 0 0 0 1 0 0 0 00 0 B06 0 0 0 0 0 0 0 0              IF NALUZ SET(ENBL.AEXP);                                      5316.000
09C3 0 2 0 0 3 0 F 0 00 0 404 0 0 0 0 0 0 0 0              T=DI, RESET(DPEFF), *JUMPD; MOVE NEW PSD2 & GO TO THIRD DECODE 5317.000
                                                 LPSD.EXIT                                                               5318.000
09C4 0 0 0 1 0 A F 0 02 0 010 0 0 0 0 0 0 0 0              I=@00010000:T;  SET RETAIN MAP BIT IN N.PSW2                   5319.000
                                                 LPSDCM.EXIT                                                             5320.000
09C5 0 4 7 0 0 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  9CF        NOD=T, *LINK EVALUATE.PSD2; GO IMPLEMENT PSD2                  5321.000
                                                 LPSD2                                                                   5322.000
09C6 0 0 0 0 4 0 F 0 19 D 280 0 0 0 0 0 0 0 0              T=R(PSW1), FETCHPC, OTHERBANK;GET PSW1 AND FETCH NEXT INST     5323.000
09C7 0 0 0 3 0 D 0 0 02 7 020 0 0 0 0 0 0 0 0              NOD=@00000002&T, I1TOI0;  TEST FOR RIGHT HALFWORD RETURN       5324.000
09C8 0 0 0 0 1 0 0 0 19 0 080 0 0 0 0 0 0 0 0              FETCHPC;  FETCH ($+4)                                         5325.000
09C9 B 5 0 0 1 0 0 0 00 C 038 0 0 0 0 0 0 0 0  938        SETRHF, IF ALUZ *GO TO RR3+2; SET RIGHT HAND FLAG              5326.000
                                                 LPSD.RH                                                                 5327.000
09CA F 3 0 0 1 0 0 0 00 7 000 0 0 0 0 0 0 0 0              I1TOI0, IF FALSE *JUMPZ;      EXIT                            5328.000
09CB 0 0 4 0 2 0 0 0 00 9 745 0 0 0 0 0 0 0 0              NOD=I0, DECODE(4), SHIFTI0, RESET(FLAG);                      5329.000
09CC 7 2 6 0 1 0 F 0 00 0 104 0 0 0 0 0 0 0 0              T=MAR, DECODE(#), IF NOFXTUNIV *JUMPD, RESET(EXFF);           5330.000
09CD 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1         *GO TO EXTG;                                                  5331.000
                                                                                                                         5332.000
                                                 ***                                                                     5333.000
                                                 *** EVALUATE PSD WORD 2                                                  5334.000
                                                 ***                                                                     5335.000
                                                 ***    ENTRY PARAMETERS :                                               5336.000
                                                 ***      T = NEW PSD 2                                                  5337.000
                                                 ***      BMUX = NEW PSD 2                                               5338.000
                                                 ***      SCRATCH(@90) = OLD PSD 2                                       5339.000
                                                 ***                                                                     5340.000
                                                                                                                         5341.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          7 5  M O D E  L P S D  I N S T R U C T I O N
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                  PANEL.EVALUATE.PSD2                                                    5342.000
09CE A 1 0 0 0 0 0 0 00 0 300 0 0 0 0 0 0 0 0                           NOD=1, IF %MODE75 *JUMPJ; EXIT IF NOT 75 MODE                    5343.000
                                                                  EVALUATE.PSD2                                                          5344.000
09CF 5 5 0 0 0 0 0 0 00 8 0E2 0 0 0 0 0 0 0 0  9E2                      NOD=1, IF %BMUX16 *GO TO REFRESH.BLK.TIMEOUT;                    5345.000
09D0 0 0 0 0 0 0 0 0 00 0 906 0 0 0 0 0 0 0 0                           NOD=1, SET(UNBLOCK); REFRESH BLOCK TIMEOUT COUNT                 5346.000
09D1 5 0 0 0 0 0 0 0 00 9 106 0 0 0 0 0 0 0 0                           NOD=T, IF %BMUX17 RESET(UNBLOCK);                                5347.000
                                                                  STORE.PSD2                                                             5348.000
09D2 0 0 0 0 0 0 8 0 00 D 000 0 0 0 0 0 0 0 0                           R(N.PSW2)=T, OTHERBANK;  STORE PSD 2                             5349.000
                                                                  EVALUATE.PSD2.MAP                                                      5350.000
09D3 0 0 0 0 0 D 0 0 02 D C00 0 0 0 0 0 0 0 0                           NOD=@C0000000&T, OTHERBANK; TEST GRANULARITY BITS                5351.000
09D4 0 0 0 0 1 0 0 0 00 2 385 0 0 0 0 0 0 0 0                           INIBK, RESET(MAPMODE);                                           5352.000
09D5 C 0 0 0 0 0 E 0 00 F B05 0 0 0 0 0 0 0 0                           FULLMAR=1(ZE), IF NALUZ SET(MAPMODE); GO MAPPED IF GRAN. NOT =0  5353.000
09D6 5 4 0 6 1 E F 0 02 8 3F9 0 0 0 0 0 0 0 0  9D9                      T=@FFFF3FFF, IF %BMUX19 *HOP USE.CURRENT.TEST.MAP;                5354.000
09D7 9 4 0 0 1 0 0 0 00 0 F7A 0 0 0 0 0 0 0 0  9DA                      IF %MAPMODE *HOP CLEAR.GRANULARITY;                              5355.000
09D8 0 6 0 0 1 0 0 0 00 0 C79 0 0 0 0 0 0 0 0  C79                      *GO TO INITIALIZE.LOAD.MAP;  GO ALTER THE MAP CONTENTS           5356.000
                                                                  USE.CURRENT.TEST.MAP                                                   5357.000
09D9 1 4 0 0 1 0 0 0 00 0 F7B 0 0 0 0 0 0 0 0  9DB                      IF MAPMODE *HOP SAVE.BPIX.CPIX;                                  5358.000
                                                                  CLEAR.GRANULARITY                                                      5359.000
09DA 0 0 0 4 0 0 F 0 02 0 3F0 0 0 0 0 0 0 0 0                           T=@3FFFFFFF&T;  SET T = @3FFF3FFF                                5360.000
                                                                  SAVE.BPIX.CPIX                                                         5361.000
09DB 0 0 0 1 1 0 1 0 02 A 900 0 0 0 0 0 0 0 0                           S=SCRATCH(@90);  GET CURRENT PSD 2                               5362.000
09DC 0 0 0 0 0 0 1 0 00 0 000 0 0 0 0 0 0 0 0                           S=S&T;  SAVE PARTS OF CURRENT PSD 2                              5363.000
09DD 0 0 0 0 1 E F 0 02 0 C00 0 0 0 0 0 0 0 0                           T=@C0000000;                                                     5364.000
09DE 0 0 0 2 0 A F 0 02 0 C00 0 0 0 0 0 0 0 0                           T=@0000C000:T;  SET T = @C000C000                                5365.000
09DF 0 0 0 4 0 0 F 0 00 0 000 0 0 0 0 0 0 0 0                           T=R(N.PSW2)&T, OTHERBANK; GET NEW GRANULARITY & INTERRUPT FLAGS  5366.000
09F0 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0                           T=S:T; MERGE OLD BPIX & CPIX WITH NEW GRAN. & INTERRUPT FLAGS    5367.000
09E1 0 1 0 1 0 0 4 0 02 A 900 0 0 0 0 0 0 0 0                           SCRATCH(@90)=T, *JUMPJ;   SAVE ALTERED NEW PSD 2                 5368.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                    7 5   M O D E   L P S D   I N S T R U C T I O N
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    REFRESH.BLK.TIMEOUT                                                    5369.000
09F2 4 5 0 0 1 0 0 0 00 8 0D2 0 0 0 0 0 0 0 0  9D2      IF UNBLOCK *GO TO STORE.PSD2;                                      5370.000
09F3 0 0 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0           SET(UNBLOCK);  CLEAR BLOCK TIMEOUT COUNT                           5371.000
09E4 0 0 0 0 1 0 0 0 00 0 106 0 0 0 0 0 0 0 0           RESET(UNBLOCK);                                                    5372.000
09E5 0 5 0 0 1 0 0 0 00 0 0D2 0 0 0 0 0 0 0 0  9D2      *GO TO STORE.PSD2;                                                 5373.000
                                                    *                                                                      5374.000
                                                    *                                                                      5375.000
                                                    *        THE PIPELINE IS ADVANCED FOR A POSSIBLE                       5376.000
                                                    *        HISTORY BIT PICKUP.                                           5377.000
                                                    *                                                                      5378.000
                                                                                                                           5379.000
                                                    *                                                                      5380.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MISCELLANEOUS INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                5381.000
                                                                                                                                5382.000
                                                                                                                                5383.000
                                                        *                                                                       5384.000
                                                        *** SECONDARY DECODE OF -HALT-, OP CODE = 00, Q=00                      5385.000
                                                        *                                                                       5386.000

                                                        (@9E8)                                                                  5387.000

                                                        HALT                                                                    5388.000

09E8 1 4 0 0 1 0 0 0 00 0 B8C 0 0 0 0 0 0 0 0  9EC              IF PRIVBIT *GO TO PRIV.ERROR5;                                  5389.000

09E9 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0                   SET(ENAUORD);                                                   5390.000

09EA 0 0 0 0 1 0 0 0 00 2 005 0 0 0 0 0 0 0 0                   HALT,RESET(ENAUORD);                                            5391.000

09EB 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829              *GO TO FETCH.RETURN;                                            5392.000

                                                        PRIV.ERROR5                                                             5393.000

09EC 0 5 0 0 1 0 0 0 00 0 0AC 0 0 0 0 0 0 0 0  9AC              *GO TO PRIV.VIOL;                                               5394.000

                                                                                                                                5395.000
                                                        *                                                                       5396.000
                                                        *** TRANSFER REGISTER TO SCRATCH PAD -TRSC-                              5397.000
                                                        *   SECONDARY DECODE OF OP CODE = 2C, Q = 0B    *                   5398.000
                                                        *                                                                       5399.000

                                                                                                                                5400.000

                                                        TRSC                                                                    5401.000

09ED 1 4 0 0 1 0 0 0 00 0 B8C 0 0 0 0 0 0 0 0  9EC              IF PRIVBIT *HOP PRIV.ERROR5;                                    5402.000

09EE 0 6 0 4 0 0 4 4 00 A 829 0 0 0 0 0 0 0 0  829              SCRATCH(R(R))=T, *GO TO FETCH.RETURN;                           5403.000

                                                                                                                                5404.000
                                                        *                                                                       5405.000
                                                        *** TRANSFER SCRATCH PAD TO REGISTER -TSCR-                              5406.000
                                                        *   SECONDARY DECODE OF OP CODE = 2C, Q = 0B    *                   5407.000
                                                        *                                                                       5408.000

                                                                                                                                5409.000

                                                        TSCR                                                                    5410.000

09EF 1 4 0 4 1 0 1 5 00 A B81 0 0 0 0 0 0 0 0  9F1              S=SCRATCH(R(S)), IF PRIVBIT *HOP PRIV.ERROR;                    5411.000

09F0 0 6 0 0 1 E 9 0 00 0 829 0 0 0 0 0 0 0 0  829              R(R)=S, *GO TO FETCH.RETURN;                                    5412.000

                                                        PRIV.ERROR                                                              5413.000

09F1 0 5 0 0 1 0 0 0 00 0 0AC 0 0 0 0 0 0 0 0  9AC              *GO TO PRIV.VIOL;                                               5414.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MISCELLANEOUS INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    WAIT.BLK.MODE.ERR                                                    5415.000

09F2 6 0 0 0 4 B B 7 02 0 200 0 0 0 0 0 0 0 0                       FR(TRACE)=%@20000000&FR(TRACE); RESET WAIT BIT       5416.000

09F3 0 0 0 0 1 0 0 0 00 0 604 0 0 0 0 0 0 0 0                       RESET(DWAIT);                                        5417.000

09F4 0 6 0 0 1 0 0 0 00 0 55B 0 0 0 0 0 0 0 0  55B                  *GO TO BLK.MODE.ERR;                                 5418.000

                                                                                                                         5419.000
                                                    *                                                                    5420.000
                                                    *** SECONDARY DECODE OF -WAIT-, OP CODE = 00, Q = 00 ***             5421.000
                                                    *                                                                    5422.000

                                                                                                                         5423.000

                                                    WAIT                                                                 5424.000

09F5 0 6 0 0 1 0 0 0 00 0 D0C 0 0 0 0 0 0 0 0  D0C                    *GOTO IPU.WAIT;                                    5425.000

                                                    *                                                                    5426.000

                                                    WAIT.RTN                                                             5427.000

                                                    *                                                                    5428.000

09F6 6 0 0 0 4 A B 7 02 0 200 0 0 0 0 0 0 0 0                       FR(TRACE)=@20000000:FR(TRACE);  SET WAIT BIT          5429.000

                                                    WAIT.LOOP                                                            5430.000

09F7 1 4 0 0 1 4 1 0 00 0 FEC 0 0 0 0 0 0 0 0  9FC                  S=S-1, IF EXTG *GO TO WAIT.EXIT; DECREMENT LOOP COUNT 5431.000

09F8 4 4 0 0 1 0 0 0 00 8 00A 0 0 0 0 0 0 0 0  9FA                  IF UNBLOCK *GO TO $+2;                               5432.000

09F9 0 4 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0  9F2                  *GO TO WAIT.BLK.MODE.ERR; GO TO BLOCK MODE TRAP      5433.000

09FA 8 4 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0  9F5                  IF ALUZ *GO TO WAIT; RELOAD LOOP COUNT IF COUNT =0   5434.000

09FB 7 4 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0  9F7                  IF NOEXTUNIV *GO TO WAIT.LOOP;  CYCLE THE LOOP       5435.000

                                                    WAIT.EXIT                                                            5436.000

09FC 5 4 0 0 1 0 0 0 00 3 00F 0 0 0 0 0 0 0 0  9FF                  IF %FFRUN *GO TO WAIT.EXIT1;                         5437.000

                                                    WAIT.INTR                                                            5438.000

09FD 6 0 0 0 4 B B 7 02 0 200 0 0 0 0 0 0 0 0                       FR(TRACE)=%@20000000&FR(TRACE); RESET WAIT BIT       5439.000

09FE 0 0 0 0 1 0 0 0 00 0 604 0 0 0 0 0 0 0 0                       RESET(DWAIT);                                        5440.000

                                                    WAIT.EXIT1                                                           5441.000

09FF 0 6 0 0 1 0 0 0 00 5 001 0 0 0 0 0 0 0 0    1                  TOGRHF, *GO TO EXTG;                                 5442.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MISCELLANEOUS INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                5443.000
                                                **                                                                              5444.000
                                                *** LOAD EFFECTIVE ADDRESS REAL INSTRUCTION, -LEAR- ***                         5445.000
                                                *** PRIMARY DECODE OF OP CODE = 80, Q = 20 ***                                  5446.000
                                                **                                                                              5447.000

                                                                                                                                5448.000
                                                LEAR                                                                            5449.000
0A00 3 6 0 0 1 E C 0 0C C 460 0 0 0 0 0 0 0 0  460      NU=S, RSTRHF, IF LATERRW *GO TO LEXTPROC;                               5450.000
0A01 0 6 7 0 1 E F 0 00 0 7F0 0 0 0 0 0 0 0 0  7F0      T=S, *LINK CHECK.CPU.MODE; GO VERIFY 75 MODE, PRIV & %EXEF              5451.000
0A02 A 4 0 0 0 0 0 0 00 0 C0A 0 0 0 0 0 0 0 0  A0A      NOD=T, IF %INDIR *GO TO LEAR.NOT.INDIRECT;                              5452.000
                                                LEAR.INDIRECT                                                                   5453.000
0A03 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0           READ, FRCWORD; FETCH INDIRECT WORD OR OPERAND                           5454.000
0A04 0 6 7 0 0 0 C 0 00 F 322 0 0 0 0 0 0 0 0  322      NU=T(ZE), *LINK DUD.WAIT.DRT; GO WAIT FOR DRT                           5455.000
0A05 3 6 0 4 3 3 7 1 00 0 45D 0 0 0 0 0 0 0 0  45D      MARIX=R(DIX)+DI, IF EXTLW *GO TO EXTPROC1;                              5456.000
0A06 3 0 0 0 1 E 0 0 02 F 01D 0 0 0 0 0 0 0 0           NOD=@01000000, IF BYTE CLDNU; SET N BIT 7 IF F BIT =1.                  5457.000
0A07 2 4 0 4 3 3 7 1 00 0 C03 0 0 0 0 0 0 0 0  A03      MARIX=R(DIX)+DI, IF INDIR *GO TO LEAR.INDIRECT;                         5458.000
0A08 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0           SET(ENAUORD);                                                           5459.000
0A09 0 4 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  A0D      *GO TO LEAR.EXIT;                                                       5460.000
                                                LEAR.NOT.INDIRECT                                                               5461.000
0A0A 0 0 0 1 2 D 0 0 02 0 080 0 0 0 0 0 0 0 0           NOD=@00080000&IO; TEST FOR F BIT IN IO                                  5462.000
0A0B 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0           SET(ENAUORD);                                                           5463.000
0A0C C 0 0 0 1 E 0 0 02 0 01D 0 0 0 0 0 0 0 0           NOD=@01000000, IF NALUZ CLDNU; SET N BIT 7 IF F BIT =1                  5464.000
                                                LEAR.EXIT                                                                       5465.000
0A0D 9 4 0 0 1 0 9 0 00 6 55F 0 0 0 0 0 0 0 0  A0F      R(R)=MAR, RDMAP, IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2;               5466.000
                                                *       GET CONTENTS OF PHYSICAL MAR AND TEST FOR MEMORY ERRORS ON LAST         5468.000
                                                *       LEVEL OF THE INDIRECT CYCLE.                                            5469.000
0A0E 0 6 7 0 1 0 0 0 00 0 2FA 0 0 0 0 0 0 0 0  2FA      *LINK CLEAR.MEM.ERROR; GO PURGE MEMORY ERRORS                           5471.000
0A0F 0 3 0 0 1 0 0 0 19 0 085 0 0 0 0 0 0 0 0           RESET(ENAUORD), FETCHPC, *JUMP;                                         5472.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU WCS DIAGOSTIC FIRMWARE
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                                                    5473.000
                                                                                                                    5475.000
                                        ***                                                                         5476.000
                                        *** THE FOLLOWING FIRMWARE IS USED WITH THE SOFTWARE WCS DIAGNOSTIC          5476.000
                                        ***                                                                         5477.000
                                        ***     NOTE:                                                               5478.000
                                        ***                                                                         5479.000
                                        ***     IF THE START OF THIS SUBROUTINE IS MOVED TO A DIFFERENT             5480.000
                                        ***     LOCATION, THE SOFTWARE DIAGNOSTIC MUST BE CHANGED.                  5481.000
                                        ***                                                                         5482.000
                                                                                                                    5483.000
                                        (@A11)                                                                      5484.000
     0000000000017F2                    WCS.ADDR        $EQ             @17F2   WCS RETURN ADDRESS                  5485.000
                                        CROM.ADDR                                                                   5486.000
0A11 0 0 0 0 1 E C 0 02 0 7F0 0 0 0 0 0 0 0 0       NU=@7F000000;   SET ITERATION COUNT FOR 7F                    5487.000
0A12 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0 0     *NOP;                                                           5488.000
0A13 6 0 5 0 1 6 1 0 00 0 000 0 0 0 0 0 0 0 0 0     S=S+1,REPEAT;   INCREMENT S TO 7F                               5489.000
0A14 0 7 0 0 1 0 0 0 00 1 7F2 0 0 0 0 0 0 0 0 17F2  *GOTO WCS.ADDR; RETURN TO WCS ADDRESS                          5490.000
```

```
                    SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                         CPU WCS INSTRUCTIONS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                        5491.000
                                                       **                                               5493.000
                                                       **  READ WRITABLE CONTROL STORAGE -RWCS-, SECONDARY DECODE (R=00)  5494.000
                                                       **                                               5495.000

                                                                                                        5496.000

                                                       (@A15)                                           5497.000

                                                       RWCS                                             5498.000

0A15 0 5 7 0 4 0 F 5 00 0 02F 0 0 0 0 0 0 0 0  A2F           T=R(S), *LINK WCS.SETUP;       GET WCS ADDRESS AND CHECK IT       5499.000

0A16 0 0 0 0 4 0 3 4 00 0 805 0 0 0 0 0 0 0 0                MAR=R(R),SET(ENAUORD);         GET MEMORY ADDRESS                 5500.000

0A17 6 0 0 0 0 0 B 3 00 0 000 0 0 0 0 0 0 0 0                FR(TEMP1)=T;                   SAVE WCS ADDRESS                   5501.000

0A18 0 0 0 0 1 0 1 0 00 8 005 0 0 0 0 0 0 0 0                S=WCS(S), RESET(ENAUORD);      READ FIRST 32 BITS FROM WCS        5502.000

0A19 0 0 0 0 1 E F 0 1F 1 885 0 0 0 0 0 0 0 0                T=S,WRITE,FRCWORD,SET(ENAUORD);     WRITE TO MEMORY 1ST WORD      5503.000

0A1A 9 4 0 0 1 0 0 0 00 0 70C 0 0 0 0 0 0 0 0  A1C           IF %OPNORESP *GOTO $+2;                                           5504.000

0A1B 0 5 0 0 1 0 0 0 00 0 021 0 0 0 0 0 0 0 0  A21           *GOTO WCS.ERROR;                                                  5505.000

0A1C 6 0 0 2 4 3 1 3 02 0 080 0 0 0 0 0 0 0 0                S=@00000800+FR(TEMP1);         COMPUTE WCS ADDRESS OF 2ND 32 BITS 5506.000

0A1D 0 0 0 0 1 0 1 0 00 8 005 0 0 0 0 0 0 0 0                S=WCS(S),RESET(ENAUORD);       READ FROM WCS                      5507.000

0A1E 0 0 0 3 1 3 3 0 02 0 040 0 0 0 0 0 0 0 0                MAR=4+MAR;                     BUMP TO NEXT MEMORY WORD           5508.000

0A1F 0 0 0 0 1 E F 0 1E 1 080 0 0 0 0 0 0 0 0                T=S,WRITE,FRCWORD;             WRITE 2ND 32 BITS TO MEMORY        5509.000

0A20 9 4 0 0 0 0 1 0 00 F 70C 0 0 0 0 0 0 0 0  A2C           S=1(ZE), IF %OPNORESP *GOTO WCS.EXIT;                             5510.000

                                                       WCS.ERROR                                        5511.000

0A21 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0                RESET(ENAUORD);                                                   5512.000

0A22 0 6 0 0 1 E C 0 02 0 530 0 0 0 0 0 0 0 0  530           NU=(CURRENT.INST.ERROR↑20&@7F000000),*GOTO CURRENT.INST.ERROR;    5513.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU CPU WCS INSTRUCTIONS

```
PC   T S M A B + D K  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5514.000
                                                **                                                                  5516.000
                                                ** WRITE WRITABLE CONTROL STORAGE -WWCS-, SECONDARY DECODE (0=00)  5517.000
                                                **                                                                  5518.000

                                                                                                                    5519.000

                                                WWCS                                                                5520.000

0A23 0 4 7 0 4 0 F 4 00 0 00F 0 0 0 0 0 0 0 0  A2F      T=R(R), *LINK WCS.SETUP;          GET WCS ADDRESS AND CHECK IT    5521.000

0A24 0 0 0 0 4 0 3 5 00 0 000 0 0 0 0 0 0 0 0           MAR=R(S);                         GET MEMORY ADDRESS              5522.000

0A25 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0           READ,FRCWORD;                     READ FIRST 32 BITS FROM MEMORY  5523.000

0A26 0 0 0 5 1 3 3 0 02 0 040 0 0 0 0 0 0 0 0           MAR=4+MAR;                        BUMP MEMORY ADDRESS             5524.000

0A27 0 0 0 0 3 0 0 0 00 0 000 0 0 0 0 0 0 0 0           BMUX=DI;                          INTERLOCK                       5525.000

0A28 0 0 0 0 1 0 0 0 1C 1 885 0 0 0 0 0 0 0 0           READ, FRCWORD, SET(ENAUORD);      READ 2ND 32 BITS                5526.000
                                                *                                         WRITE FIRST 32 BITS TO WCS      5526.000
                                                *                                         AND CHECK FOR ERROR ON 1ST READ 5529.000

0A29 1 4 0 0 3 0 0 0 00 7 551 0 0 0 0 0 0 0 0  A21      WCS(S)=DI,IF OPNORESP:OPTIMEOUT:OPRNOPE *GOTO WCS.ERROR;  5531.000

0A2A 0 0 0 2 0 3 1 0 02 0 080 0 0 0 0 0 0 0 0           S=@00000800+T;                    BUMP WCS ADDRESS                5532.000

0A2B 0 0 0 0 3 0 0 0 00 7 005 0 0 0 0 0 0 0 0           WCS(S)=DI, RESET(ENAUORD);        WRITE 2ND 32 BITS TO WCS        5533.000

                                                WCS.EXIT                                                            5534.000

0A2C 9 4 0 0 1 0 0 0 00 0 0EF 0 0 0 0 0 0 0 0  A2E      IF %FLAG *NOP $+2;                RESET RH FLAG STATUS            5535.000

0A2D 0 0 0 0 1 0 0 0 00 C 705 0 0 0 0 0 0 0 0           SETPHF, RESET(FLAG);                                      5536.000

0A2E 0 6 1 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829      SETCC(#), *GOTO FETCH.RETURN;                             5537.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU CPU WCS INSTRUCTIONS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                5538.000

                                                WCS.SETUP                                                       5539.000

0A2F 0 0 0 0 1 0 0 0 00 0 705 0 0 0 0 0 0 0 0            RESET(FLAG);                SAVE CURRENT RH FLAG STATUS    5540.000

0A30 9 4 0 0 1 0 0 0 00 0 882 0 0 0 0 0 0 0 0  A32      IF %PRIVBIT *HOP $+2;                              5541.000

0A31 0 6 0 0 1 0 0 0 00 0 9AC 0 0 0 0 0 0 0 0  9AC      *GO TO PRIV.VIOL;EXIT IF NOT PRIVLEGED             5542.000

0A32 4 4 0 6 1 6 1 0 02 1 FF9 0 0 0 0 0 0 0 0  A39      S=@FFFFEFFF+1, IF %WCS *HOP WCS.NOT.PRESENT; S = @FFFFF000  5543.000

0A33 A 4 0 0 0 D 0 0 00 0 A05 0 0 0 0 0 0 0 0  A35      NOD=S&T, IF %RHFLAG *GOTO $+2;                     5544.000

0A34 0 0 0 0 0 D 0 0 0C 0 F05 0 0 0 0 0 0 0 0            NOD=S&T, RSTRHF, SET(FLAG); REMEMBER RIGHT HAND FLAG  5545.000

0A35 0 0 0 2 0 D 1 0 02 0 080 0 0 0 0 0 0 0 0            S=@00000800&T;              =0 IF WCS ADDRESS 0-7FF    5546.000

0A36 B 1 0 0 0 3 F 0 00 6 000 0 0 0 0 0 0 0 0            T=S+T,SDES1, IF ALU7 *JUMPJ;  ADJUST WCS ADDRESS IF 800-FFF  5547.000

                                              * FALL THROUGH IF WCS ADDRESS OUT OF RANGE ( 000 - FFF )          5548.000

                                                WCS.ADDR.OUT.OF.RANGE                                           5549.000

0A37 0 0 0 0 1 E 1 0 02 0 400 0 0 0 0 0 0 0 0            S=@40000000;     SET CC 1 FOR WCS ADDRESS ERROR.    5550.000

0A38 0 5 0 0 1 0 0 0 00 0 02C 0 0 0 0 0 0 0 0  A2C      *GO TO WCS.EXIT;                                   5551.000

                                                WCS.NOT.PRESENT                                                 5552.000

0A39 0 0 0 0 1 E 1 0 02 0 400 0 0 0 0 0 0 0 0            S=@40000000;            SET CC1 FOR NO WCS OR ADDRESS ERROR  5553.000

0A3A 0 0 0 0 1 0 0 0 00 0 501 0 0 0 0 0 0 0 0            SETCC(S);   SET CC 1                               5554.000

0A3B 0 6 0 0 1 0 0 0 00 0 9A9 0 0 0 0 0 0 0 0  9A9      *GO TO UNDEF;  GO TO UNDEFINED INSTRUCTION TRAP     5555.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU CPU WCS INSTRUCTIONS
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                                                                    5556.000

                                                    (@A3E)                                                          5557.000

                                                    ENTER.WCS                                                       5558.000

0A3E 6 0 1 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPS;                    BRANCH TO THE WCS ADDRESS IN THE S-REG 5559.000
                                                    **                                                              5561.000
                                                    **  JUMP TO WRITABLE CONTROL STORAGE -JWCS-, SECONDARY DECODE (Q=3F)     5562.000
                                                    **                                                              5563.000

                                                                                                                    5564.000

                                                    JWCS                                                            5565.000

0A3F 4 4 0 3 1 D F 0 02 1 3F9 0 0 0 0 0 0 0 0  A49      I=@03F&MAR, IF %WCS *GOTO WCS.NOT.PRESENT1; GET 6-BIT JUMP ADDR  5566.000

                                                    JWCS.1                                                          5567.000

0A40 A 4 0 0 1 0 0 0 00 0 C05 0 0 0 0 0 0 0 0  A45      IF %INDIR *GO TO JUMP.WCS;                                  5568.000

0A41 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B      *LINK MEMORY.READ;         GO FETCH INDIRECT WORD (JUMP ADDRESS)  5569.000

0A42 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0           MARIX=R(DIX)+DI;           INDEX INDIRECT OPERAND              5570.000

0A43 5 4 0 3 1 D F 0 02 0 3F0 0 0 0 0 0 0 0 0  A40      I=@03F&MAR, IF %EXTLW *GOTO JWCS.1; GET 6-BIT JUMP ADDR      5571.000

0A44 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      *GOTO EXIPROC1;                                             5572.000

                                                    JUMP.WCS                                                        5573.000

0A45 0 0 0 2 0 A 1 0 02 0 100 0 0 0 0 0 0 0 0           S=@00001000:I;  GENERATE WCS JUMP ADDRESS                   5574.000

0A46 0 5 7 0 1 0 0 0 00 0 03E 0 0 0 0 0 0 0 0  A3E      PUSHJ, *GOTO ENTER.WCS;     SAVE RETURN ADDRESS IN J STACK  5575.000

                                                    WCS.RETURN.EXIT                                                 5576.000

0A47 0 0 0 0 1 0 0 0 1D 0 084 0 0 0 0 0 0 0 0           FETCHPC, RESET(HIREG);                                      5577.000

0A48 0 3 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0           *JUMPZ, RESET(ENAUORD);                                     5578.000

                                                    WCS.NOT.PRESENT1                                                5579.000

0A49 0 5 0 0 1 0 0 0 00 0 039 0 0 0 0 0 0 0 0  A39      *GO TO WCS.NOT.PRESENT;                                     5580.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU WCS SYSTEM RESET
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                    5581.000
                                                    ***                                                             5583.000
                                                    ***  WCS SYSTEM RESET                                           5584.000
                                                    ***                                                             5585.000
                                                    ***    THIS SUB-ROUTINE INITIALIZES THE FIRST 64 64-BIT WCS LOCATIONS   5586.000
                                                    ***    TO A 13-BIT BRANCH TO UNDEFINED INSTRUCTION.             5587.000
                                                    ***                                                             5588.000
                                                                                                                    5589.000
                                                                                                                    5590.000
                                                    NOT.WCS.EXIT.SYS.RESET                                          5591.000
0A4A 4 1 0 0 1 0 0 0 00 1 B04 0 0 0 0 0 0 0 0             SET(TRACEFF), IF %WCS *JUMPJ; EXIT SYS RESET IF NO WCS PRESET   5592.000
0A4B 0 0 0 0 0 0 0 0 00 0 805 0 0 0 0 0 0 0 0             NOD=1, SET(ENAUORD);                                  5593.000
0A4C 0 4 7 0 0 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  A4E        NOD=1, *LINK WCS.SYS.RESET;                           5594.000
0A4D 0 1 0 0 1 0 0 0 00 0 B04 0 0 0 0 0 0 0 0             SET(TRACEFF),*JUMPJ;*RETURN TO SYSTEM RESET           5595.000
                                                    WCS.SYS.RESET                                                   5596.000
0A4E 0 0 0 0 1 E C 0 02 0 3F7 0 0 0 0 0 0 0 0             NU=@3F000000, CLRS; SET ITERATION COUNT TO 64 AND INITIAL ADDR =0 597.000
0A4F 0 4 7 0 1 E 6 0 02 0 0E5 0 0 0 0 0 0 0 0  A55        DI=@0F000000, *LINK LOAD.WCS; SET MOST SIGNIFICANT 32-BITS FOR A 5598.000
                                                    *             BRANCH 13  (S FIELD = @7).                        5600.000
0A50 0 0 0 0 1 E 6 0 02 0 090 0 0 0 0 0 0 0 0             DI=(UNDEF↑16&@0F000000); LOAD MOST SIGNIFICANT DIGIT OF 'UNDEF' 5602.000
                                                    *             ADDRESS INTO DI BITS 4-7  (P FIELD).              5604.000
0A51 0 0 0 1 5 A 6 0 02 0 A90 0 0 0 0 0 0 0 0             DI=(UNDEF↑16&@00FF0000) : DI; LOAD LEAST SIGNIFICANT 2 DIGITS OF 5606.000
                                                    *             'UNDEF' ADDRESS INTO DI BITS 8-15 (C AND H FIELDS). THIS 5608.000
                                                    *             FORMS A 13-BIT BRANCH ADDRESS IN X,P,C, AND H FIELD OF  5609.000
                                                    *             OF THE LEAST SIGNIFICANT 32-BIT WCS WORD.         5610.000
0A52 0 0 0 0 1 E C 0 02 0 3F0 0 0 0 0 0 0 0 0             NU=@3F000000;  SET ITERATION COUNT FOR 64             5612.000
0A53 0 4 7 2 1 E 1 0 02 0 085 0 0 0 0 0 0 0 0  A55        S=@0800, *LINK LOAD.WCS; SET WCS ADDR FOR LEAST SIGNIF 32-BITS  5613.000
0A54 0 1 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0             RESET(ENAUORD), *JUMPJ;  EXIT INITIALIZE WCS          5614.000
                                                                                                                    5615.000
                                                    LOAD.WCS                                                        5616.000
0A55 0 0 0 0 3 0 0 0 00 7 000 0 0 0 0 0 0 0 0             WCS(S)=DI;  LOAD DI INTO WCS ADDRESSED BY S REG.       5617.000
0A56 A 4 3 0 1 6 1 0 00 0 905 0 0 0 0 0 0 0 0  A55        S=S+1, DECRN, IF %NCTRZ *GO TO LOAD.WCS; INCREMENT WCS ADDR AND  5618.000
                                                    *             DECREMENT ITERATION COUNT.                        5620.000
0A57 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *JUMPJ;  EXIT WCS LOAD                                5622.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU SERIAL PANEL
PC TSMAB+DR YX PCH+BDM+BAD ADDR

```
                                                                        5623.000
;*********************************************************************  5625.000
;***    SERIAL PANEL FIRMWARE 11 JULY 1978  (WH&SP2)               ***  5626.000
;*********************************************************************  5627.000

                                                                        5628.000
;*********************************************************************  5629.000
;**                                                                 **  5630.000
;**                                                                 **  5631.000
;**     THIS IS A DESCRIPTION OF THE CONTENTS OF SCRATCH PAD        **  5632.000
;**                                                                 **  5633.000
;**                                                                 **  5634.000
;**     SCRATCH PAD LOCATION          CONTENTS                      **  5635.000
;**                                                                 **  5636.000
;**             86                    BITS00-07=TRANSMIT COUNT      **  5637.000
;**             86                    BITS08-11=FLAG                **  5638.000
;**             86                    BITS24-27=ID                  **  5639.000
;**             86                    BITS28-31=OUTPUT DATA         **  5640.000
;**                                                                 **  5641.000
;**             87                    BITS00-31=TRANSMIT DATA       **  5642.000
;**                                                                 **  5643.000
;**             88                    BITS20-31=DISPLAY 'C'         **  5644.000
;**                                                                 **  5645.000
;**             89                    BITS00-31=DISPLAY 'A'         **  5646.000
;**                                                                 **  5647.000
;**             8A                    BITS20-31=DISPLAY 'D'         **  5648.000
;**                                                                 **  5649.000
;**             8B                    BITS00-31=DISPLAY 'B'         **  5650.000
;**                                                                 **  5651.000
;**             8C                    BITS04-07=FUNCTION            **  5652.000
;**                                                                 **  5653.000
;**             8D                    ADDRESS STOP                  **  5654.000
;**                                                                 **  5655.000
;*********************************************************************  5656.000
;**  THIS DESCRIBES THE ENCODING OF PLNDATA, C-DISP, & D-DISP BITS  **  5657.000
;**                                                                 **  5658.000
;******* PNLDATA BITS 4-7 ********** HEX ****************************  5659.000
;**                                                                 **  5660.000
;**      INCREMENT & READ             1   NOTE: FUNCTION IF         **  5661.000
;**      WRITE & INCREMENT            2         BIT 4=0             **  5662.000
;**      INSTRUCTION STEP             3                             **  5663.000
;**      INSTRUCTION STOP             4                             **  5664.000
;**      OPERAND READ STOP            5                             **  5665.000
;**      OPERAND WRITE STOP           6                             **  5666.000
;**      PANEL UART ERROR             7                             **  5667.000
;**                                                                 **  5668.000
;******* PNLDATA BITS 4-7 ********** HEX ****************************  5669.000
;**                                                                 **  5670.000
;**      KEYBOARD                     8   NOTE: DATA  IF            **  5671.000
;**      EXTENDED FUNCTION            9   BIT 4=1                   **  5672.000
;**      READ                         A                             **  5673.000
;**      WRITE                        B                             **  5674.000
;**                                                                 **  5675.000
;*********************************************************************  5676.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U S E R I A L P A N E L
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
**    NOTE: FOR PNLDATA BIT00=1 DEFINES FUNCTION IN BITS4-7        **    5677.000
**                          BIT00=0 DEFINES DATA IN BITS4-7        **    5678.000
*************************************************************************5679.000
**                                                                     **    5680.000
**                                                                     **    5681.000
**                                                                     **    5682.000
**                                                                     **    5683.000
******* DISPLAY 'C' *************** HEX ****************************** *    5684.000
**                                                                     **    5685.000
**      MEMORY ADDRESS              001                                **    5686.000
**      PROGRAM STATUS WORD         002                                **    5687.000
**      PROGRAM COUNTER             004                                **    5688.000
**      OPERATOR FAULT              008                                **    5689.000
**      EFFECTIVE ADDRESS           010                                **    5690.000
**      MEMORY DATA                 020                                **    5691.000
**      INSTRUCTION                 040                                **    5692.000
**      STOP                        080                                **    5693.000
**      INSTRUCTION STOP            100                                **    5694.000
**      OPERAND READ STOP           200                                **    5695.000
**      OPERAND WRITE STOP          400                                **    5696.000
**      KEYBOARD                    800                                **    5697.000
**                                                                     **    5698.000
*************************************************************************5699.000

                                                                            5700.000
*************************** DISPLAY 'D' ******************************** *    5701.000
**                                                                     **    5702.000
**            BIT 19   =EXTENDED FUNCTION 1                            **    5703.000
**            BITS20-23=REGISTER 'A' (EVEN)                            **    5704.000
**            BITS24-27=REGISTER 'B' (ODD)                             **    5705.000
**            BIT28=ENABLE REGISTER 'A'                                **    5706.000
**            BIT29=ENABLE REGISTER 'B'                                **    5707.000
**            BIT30=CONTROL SWITCHES                                   **    5708.000
**            BIT31=ERROR                                              **    5709.000
**                                                                     **    5710.000
*************************************************************************5711.000

                                                                            5712.000
***************************** TC.FLG.ID. ******************************* *    5713.000
**                                                                     **    5714.000
**            BITS00-07=TRANSMIT COUNT / TC.                           **    5715.000
**            BITS08-11=FLAG / FLG.                                    **    5716.000
**            BITS24-27=IDENTIFICATION / ID.                           **    5717.000
**            BITS28-31=OUTPUT DATA                                    **    5718.000
**                                                                     **    5719.000
*************************************************************************5720.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU SERIAL PANEL
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                5721.000
                                                                                                                                5722.000
                                                        (@A58)                                                                  5723.000
                                                        S.P.INC.PC                                                              5724.000
0A58 0 5 7 0 1 0 0 0 00 0 07A 0 0 0 0 0 0 0 0  A7A              *LINK S.P.FETCH.C;      GET C DISPLAY IN T REG                  5725.000
0A59 0 0 0 3 0 0 0 0 02 0 010 0 0 0 0 0 0 0 0                   NOD=@00000001&T;  TEST FOR MA BIT IN 'C'                      5726.000
0A5A 0 0 0 0 4 0 F 0 00 0 800 0 0 0 0 0 0 0 0                   I=R(PCMASK), OTHERBANK;   LOAD MASK @0007FFFC INTO T REG        5727.000
0A5B C 6 0 0 1 0 0 0 00 0 C0C 0 0 0 0 0 0 0 0  C0C              IF NALU7 *GO TO S.P.SET.ALL.FLAGS;                              5728.000
0A5C 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B              IF FERUN *GO TO S.P.FAULT.2;                                    5729.000
0A5D 0 0 0 1 1 0 1 0 02 A 890 0 0 0 0 0 0 0 0                   S=SCRATCH(@89);               DISPLAY A (PC OR PSW)             5730.000
0A5E 0 6 0 0 0 0 2 0 00 0 C0C 0 0 0 0 0 0 0 0  C0C              PC=S&T, *GO TO S.P.SET.ALL.FLAGS;                               5731.000
                                                        ** SERIAL.PANEL.ATTN                                                    5732.000
                                                        S.P.ATTN                                                                5733.000
0A5F 0 0 0 0 1 0 0 0 00 0 804 0 0 0 0 0 0 0 0                   SFT(HIREG);                                                     5734.000
0A60 1 4 0 0 1 0 0 0 00 0 F12 0 0 0 0 0 0 0 0  A62              IF UARTDAV *GO TO UART.INPUT;                                   5735.000
0A61 0 5 0 0 1 0 0 0 00 0 06D 0 0 0 0 0 0 0 0  A8D              *GO TO S.P.UART.TRMT; IF DATA NOT AVAILABLE CHECK TRMT          5736.000
                                                        UART.INPUT                                                              5737.000
0A62 9 4 0 0 1 0 0 0 00 0 F04 0 0 0 0 0 0 0 0  A64              IF %UARTERR *GO TO $+2;                                         5738.000
0A63 0 6 0 0 1 0 0 0 00 0 C48 0 0 0 0 0 0 0 0  C48              *GO TO S.P.ERROR.1; ERROR IN UART                               5739.000
                                                        **                                                                      5740.000
                                                        **        TURN OFF EXTENDED FCN 1 BIT IN D-DISPLAY. THIS DEMANDS THAT   5741.000
                                                        **        ANY NEW KEYIN SET D-DISPLAY PENDING TO ACTUALLY TURN OFF THE  5742.000
                                                        **        SERIAL PANEL TIMER.                                           5743.000
                                                        **                                                                      5744.000
0A64 0 5 7 0 1 0 0 0 00 0 07B 0 0 0 0 0 0 0 0  A7B              *LINK S.P.FETCH.D;                                              5745.000
0A65 0 0 0 6 0 0 F 0 02 0 EF0 0 0 0 0 0 0 0 0                   T=@FFFFEFFF&T;            TURN OFF BIT 19                       5746.000
0A66 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0                   SCRATCH(@8A)=T;                                                 5747.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    S F R I A L   P A N E L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0A67 0 0 0 0 6 0 0 0 00 0 A05 0 0 0 0 0 0 0 0          NOD=PNLDATA, SET(UARTDAV);                                    5748.000

0A68 5 4 0 0 6 D F 0 02 4 0FA 0 0 0 0 0 0 0 0  A6A     T=@0F000000&PNLDATA, IF %BMUX00 *HOP $+2;                     5749.000

0A69 0 5 0 0 1 0 0 0 00 0 083 0 0 0 0 0 0 0 0  A83     *GO TO S.P.DATA; DATA IN BITS 4-7                             5750.000

0A6A 0 5 7 0 1 0 0 0 00 0 07F 0 0 0 0 0 0 0 0  A7F     *LINK S.P.KEYBOARD.OFF;                                       5751.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FUNCTION DATA

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    5752.000
                                                        **                                                          5753.000
                                                        **      ENTRY PARAMETERS:  T=@0F000000&PNLDATA              5754.000
                                                        **                                                          5755.000
                                                        S.P.FUNCTION.DATA                                           5756.000
0A6B 0 0 0 1 0 0 4 0 02 A 8C0 0 0 0 0 0 0 0 0           SCRATCH(@8C)=T;         SAVE FUNCTION                       5757.000
0A6C 0 0 0 0 1 0 0 0 00 2 000 0 0 0 0 0 0 0 0           INIBL;                                                      5758.000
0A6D 0 0 0 3 0 E 1 0 02 0 A70 0 0 0 0 0 0 0 0           BMUX=T, S=(S.P.JUMP.TBL↑-4&@00FF); 'GENERATE BASE ADDR OF TBL  5759.000
0A6E 5 6 0 0 1 0 0 6 00 4 8C2 0 0 0 0 0 0 0 0  8C2      IF %BMUX00 *GO TO S.P.KEYBOARD;                             5760.000
0A6F 0 6 7 2 1 E 1 0 00 0 869 0 0 0 0 0 0 0 0  869      S=SNIBL, *LINK DUD.86X; SHIFT BASE ADDR LEFT AND MERGE VECTOR  5761.000
0A70 6 0 1 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPS;                                                     5762.000
                                                        **                                                          5763.000
                                                        **      LOCATION CRITICAL JUMP TABLE - CANNOT BE MOVED!     5764.000
                                                        **                                                          5765.000
                                                        (@A71)                                                      5766.000
                                                        S.P.JUMP.TBL                                                5767.000
0A71 0 5 0 0 1 0 0 0 00 0 0E1 0 0 0 0 0 0 0 0  AE1      *GO TO S.P.INC.RD;                                          5768.000
0A72 0 5 0 0 1 0 0 0 00 0 0C3 0 0 0 0 0 0 0 0  AC3      *GO TO S.P.WRT.INC;                                         5769.000
0A73 0 6 0 0 1 0 0 0 00 0 8D5 0 0 0 0 0 0 0 0  8D5      *GO TO S.P.INST.STEP;                                       5770.000
0A74 0 6 0 0 1 0 0 0 00 0 8DF 0 0 0 0 0 0 0 0  8DF      *GO TO S.P.INST.STOP;                                       5771.000
0A75 0 6 0 0 1 0 0 0 00 0 8E5 0 0 0 0 0 0 0 0  8E5      *GO TO S.P.OPRND.RD.STOP;                                   5772.000
0A76 0 6 0 0 1 0 0 0 00 0 8F7 0 0 0 0 0 0 0 0  8F7      *GO TO S.P.OPRND.WRT.STOP;                                  5773.000
0A77 0 6 0 0 1 0 0 0 00 0 C4A 0 0 0 0 0 0 0 0  C4A      *GO TO S.P.ERROR.2;                                         5774.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          F E T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5775.000
                                                    S.P.FETCH.A                                                     5776.000
0A78 0 4 0 1 1 0 1 0 02 A 89F 0 0 0 0 0 0 0 0  A7E      S=SCRATCH(@89), *HOP S.TO.T.DUD;                           5777.000
                                                    S.P.FETCH.B                                                     5778.000
0A79 0 4 0 1 1 0 1 0 02 A 8BE 0 0 0 0 0 0 0 0  A7E      S=SCRATCH(@8B), *HOP S.TO.T.DUD;                           5779.000
                                                    S.P.FETCH.C                                                     5780.000
0A7A 0 4 0 1 1 0 1 0 02 A 88E 0 0 0 0 0 0 0 0  A7E      S=SCRATCH(@88), *HOP S.TO.T.DUD;                           5781.000
                                                    S.P.FETCH.D                                                     5782.000
0A7B 0 4 0 1 1 0 1 0 02 A 8AE 0 0 0 0 0 0 0 0  A7E      S=SCRATCH(@8A), *HOP S.TO.T.DUD;                           5783.000
                                                    S.P.FETCH.FUNCTION                                              5784.000
0A7C 0 4 0 1 1 0 1 0 02 A 8CE 0 0 0 0 0 0 0 0  A7E      S=SCRATCH(@8C), *HOP S.TO.T.DUD;                           5785.000
                                                    S.P.FETCH.TC.FLG.ID                                             5786.000
0A7D 0 0 0 1 1 0 1 0 02 A 860 0 0 0 0 0 0 0 0           S=SCRATCH(@86);                                            5787.000
                                                    S.TO.T.DUD                                                      5788.000
0A7E 0 1 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0           T=S, *JUMPJ;                                               5789.000
                                                    S.P.KEYBOARD.OFF                                                5790.000
0A7F 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0           S=SCRATCH(@88);              DISPLAY-C INTO S-REG           5791.000
0A80 0 0 0 6 1 E 6 0 02 0 F70 0 0 0 0 0 0 0 0           DI=@FFFFF7FF;              KEYBOARD MASK INTO FILE          5792.000
0A81 0 0 0 0 3 D 6 0 00 0 000 0 0 0 0 0 0 0 0           DT=S&DI;                   MASK                             5793.000
0A82 0 1 0 1 3 0 4 0 02 A 880 0 0 0 0 0 0 0 0           SCRATCH(@88)=DI, *JUMPJ;       RESTORE C-DISPLAY            5794.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  CPU                                                                          D A T A
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                                5795.000

                                                                S.P.DATA                                                        5796.000

0A83 0 5 7 0 1 0 0 0 00 0 07C 0 0 0 0 0 0 0 0  A7C        *LINK S.P.FETCH.FUNCTION;                                            5797.000

0A84 0 4 7 0 0 8 0 0 02 0 08C 0 0 0 0 0 0 0 0  A8C        NOD=@08000000!T, *LINK DUD.A8Y; TEST FOR KEYBOARD DATA               5798.000

0A85 8 6 0 0 1 0 0 0 00 0 B95 0 0 0 0 0 0 0 0  B95        IF ALUZ *GO TO S.P.KBDATA;                                           5799.000

0A86 0 5 7 0 1 0 0 0 00 0 07F 0 0 0 0 0 0 0 0  A7F        *LINK S.P.KEYBOARD.OFF; NOT KEYBOARD SO TURN OFF LITE                5800.000

0A87 0 4 7 0 0 8 0 0 02 0 09C 0 0 0 0 0 0 0 0  A8C        NOD=@09000000!T, *LINK DUD.A8X; TEST FOR EXTENDED FUNCTION           5801.000

0A88 8 6 0 0 1 0 0 0 00 0 BFE 0 0 0 0 0 0 0 0  BFE        IF ALUZ *GO TO S.P.EXTFUNC;                                          5802.000

0A89 0 4 7 0 0 8 0 0 02 0 0CC 0 0 0 0 0 0 0 0  A8C        NOD=@0C000000&T, *LINK DUD.A8X;                                      5803.000

0A8A 8 6 0 0 1 0 0 0 00 0 C4A 0 0 0 0 0 0 0 0  C4A        IF ALUZ *GO TO S.P.ERROR.2;                                          5804.000

0A8B 0 5 0 0 1 0 0 0 00 0 0F9 0 0 0 0 0 0 0 0  AF9        *GO TO S.P.PD.OR.PT;                                                 5805.000

                                                                DUD.A8X                                                         5806.000

0A8C 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPJ;                                                              5807.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U U A R T . T B M T

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                5808.000
                                                        S.P.UART.TBMT                                           5809.000
0A8D 0 5 7 0 1 0 0 0 00 0 09B 0 0 0 0 0 0 0 0  A9B          *LINK S.P.S.TO.FULLMAR;                             5810.000
0A8E 0 0 0 3 1 E F 0 00 0 000 0 0 0 0 0 0 0 0               T=SNIBR; PLACES FLAG BITS IN BITS 16-19             5811.000
0A8F A 5 0 0 0 0 0 0 00 0 9AC 0 0 0 0 0 0 0 0  AAC          NOD=T,IF %NCTRZ *GO TO S.P.XMIT; TRANSMIT COUNT TEST 5812.000
                                                        S.P.PEND.TEST                                           5813.000
0A90 5 4 0 0 0 0 0 0 00 8 00E 0 0 0 0 0 0 0 0  A9E          NOD=T, IF %BMUX16 *GO TO S.P.RESET.DISPA.PNDG;       5814.000
0A91 5 5 0 0 0 0 0 0 00 9 0A2 0 0 0 0 0 0 0 0  AA2          NOD=1, IF %BMUX17 *GO TO S.P.RESET.DISPB.PNDG;       5815.000
0A92 5 5 0 0 0 0 0 0 00 A 0B6 0 0 0 0 0 0 0 0  AB6          NOD=T, IF %BMUX18 *GO TO S.P.RESET.DISPC.PNDG;       5816.000
0A93 5 5 0 0 1 0 0 0 00 B 0BE 0 0 0 0 0 0 0 0  ABE          IF %BMUX19 *GO TO S.P.RESET.DISPD.PNDG;             5817.000
0A94 0 0 0 1 1 0 1 0 02 A 8D0 0 0 0 0 0 0 0 0               S=SCRATCH(@8D); FETCH ADDRESS STOP BITS FOR TESTING 5818.000
0A95 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0               T=S;  TRANSFER ADDRESS STOP BITS TO T REG           5819.000
0A96 0 0 0 7 0 3 F 0 00 0 EF0 0 0 0 0 0 0 0 0               T=@FFFFFFEF+T; TEST FOR ADDRESS STOP (T=00000011)   5820.000
0A97 0 0 0 1 0 0 4 0 02 A 8D0 0 0 0 0 0 0 0 0               SCRATCH(@8D)=T; CLEAR ADDRESS STOP IF RESULTS OF T EQUAL ZERO 5821.000
0A98 B 6 0 0 1 0 0 0 00 0 C31 0 0 0 0 0 0 0 0  C31          IF ALUZ *GO TO S.P.ADDRESS.STOP;                    5822.000
0A99 0 0 0 0 1 0 0 0 00 0 405 0 0 0 0 0 0 0 0               RESET(ENATBMT);  DISABLE TBMT WHEN ALL OUTPUTS ARE COMPLETE 5823.000
0A9A 0 6 0 0 1 0 0 0 00 0 C19 0 0 0 0 0 0 0 0  C19          *GO TO EXIT;                                        5824.000
                                                        S.P.S.TO.FULLMAR                                        5825.000
0A9B 0 0 0 1 1 0 1 0 02 A 860 0 0 0 0 0 0 0 0               S=SCRATCH(@86); FETCH TC.FLG.ID                     5826.000
0A9C 0 0 0 0 1 E E 0 00 0 00D 0 0 0 0 0 0 0 0               FULLMAR=S, CLDNU;                                   5827.000
0A9D 0 1 0 3 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0               S=SNIBR, *JUMPJ;                                    5828.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5829.000
                                                    S.P.RESET.DISPA.PNDG                                            5830.000
0A9E 0 5 7 0 1 0 0 0 00 0 07D 0 0 0 0 0 0 0 0  A7D      *LINK S.P.FETCH.TC.FLG.ID;                                  5831.000
0A9F 0 0 0 1 0 D F 0 02 0 700 0 0 0 0 0 0 0 0           T=@00700000&T;  RESET DISP 'A' PEND AND SET 'ID' FOR 'A'     5832.000
0AA0 0 0 0 1 1 0 1 0 02 A 890 0 0 0 0 0 0 0 0           S=SCRATCH(@89); FETCH 'A'                                   5833.000
                                                    S.P.INSERT.ABTB                                                 5834.000
0AA1 0 4 0 0 1 E 6 0 00 0 006 0 0 0 0 0 0 0 0  AA6      DI=S,*GO TO S.P.TRNMT.COUNT.ADDR;                           5835.000
                                                    S.P.RESET.DISPB.PNDG                                            5836.000
0AA2 0 5 7 0 1 0 0 0 00 0 07D 0 0 0 0 0 0 0 0  A7D      *LINK S.P.FETCH.TC.FLG.ID;                                  5837.000
0AA3 0 0 0 1 0 D F 0 02 0 B00 0 0 0 0 0 0 0 0           T=@00B00000&T;  RESET DISP. 'B' PENDING AND CLEAR 'ID'      5838.000
0AA4 0 0 0 3 0 A F 0 02 0 100 0 0 0 0 0 0 0 0           T=@00000010:T;  SET 'ID' FOR 'B'                            5839.000
0AA5 0 4 0 1 1 0 1 0 02 A 8B1 0 0 0 0 0 0 0 0  AA1      S=SCRATCH(@8B),*GO TO S.P.INSERT.ABTB; FETCH 'B'             5840.000
                                                    S.P.TRNMT.COUNT.ADDR                                            5841.000
0AA6 0 0 0 1 3 0 4 0 02 A 870 0 0 0 0 0 0 0 0           SCRATCH(@87)=DI; LOAD TRANSMIT BUFFER                       5842.000
0AA7 0 4 0 0 0 A F 0 02 0 089 0 0 0 0 0 0 0 0  AA9      T=@08000000:T, *GO TO $+2; SET TRANSMIT COUNT FOR 'A' AND 'B'  5843.000
0AA8 0 0 0 0 3 A F 0 02 0 040 0 0 0 0 0 0 0 0           T=@04000000:DI; SET TRANSMIT COUNT FOR 'C' AND 'D'           5844.000
0AA9 0 0 0 1 0 0 4 0 02 A 860 0 0 0 0 0 0 0 0           SCRATCH(@86)=T; STORE TC.FLG.ID                             5845.000
0AAA 0 0 0 0 1 0 0 0 00 0 C05 0 0 0 0 0 0 0 0           SET(ENATBMT);  ENABLE THE TRANSMIT BUFFER EMPTY LOGIC       5846.000
0AAB 0 6 7 0 1 0 0 0 00 0 309 0 0 0 0 0 0 0 0  309      *LINK TRIPLE.DUD;  ALLOW 4 CLOCK FOR TBMT TO ACTIVATE       5847.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU TRANSMIT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5848.000
                                                S.P.XMIT                                                            5849.000
0AAC 0 0 0 1 1 0 1 0 02 A 870 0 0 0 0 0 0 0 0           S=SCRATCH(@87);   FETCH TRANSMIT DATA                     5850.000
0AAD 1 4 0 0 1 E 6 0 00 0 D9F 0 0 0 0 0 0 0 0  AAF      DI=S, IF UARTTBMT *HOP UART.XMIT; MOVE XMIT DATA TO DI      5851.000
                                                XMIT.EXIT                                                           5852.000
0AAE 0 6 0 0 1 0 0 0 00 0 C19 0 0 0 0 0 0 0 0  C19      *GO TO EXIT;   EXIT IF NOT TBMT                           5853.000
                                                UART.XMIT                                                           5854.000
0AAF 0 5 7 0 1 0 0 0 00 0 09B 0 0 0 0 0 0 0 0  A9B      *LINK S.P.S.TO.FULLMAR; GO FETCH XMIT COUNT AND FLAGS      5855.000
0AB0 0 0 3 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0           T=DI, DECRN; DECREMENT THE TRANSMIT COUNT                   5856.000
0AB1 0 0 0 2 1 E 1 0 00 2 905 0 0 0 0 0 0 0 0           S=SNIBL,TNIBL, SET(UARTDS);                                 5857.000
0AB2 0 0 0 1 1 0 4 0 02 A 860 0 0 0 0 0 0 0 0           SCRATCH(@86)=MAR; SAVE UPDATE XMIT COUNT AND FLAGS          5858.000
0AB3 0 0 0 0 1 0 0 0 00 0 105 0 0 0 0 0 0 0 0           RESET(UARTDS);                                              5859.000
0AB4 0 0 0 1 0 0 4 0 02 A 870 0 0 0 0 0 0 0 0           SCRATCH(@87)=T;          SAVE UPDATED TRANSMIT DATA         5860.000
0AB5 0 6 0 0 1 0 0 0 00 0 C19 0 0 0 0 0 0 0 0  C19      *GO TO EXIT;                                                5861.000
                                                                                                                    5862.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5863.000
                                                S.P.RESET.DISPC.PNDG                                                5864.000
0AB6 0 5 7 0 1 0 0 0 00 0 07D 0 0 0 0 0 0 0 0  A7D        *LINK S.P.FETCH.TC.FLG.ID;                               5865.000
0AB7 0 0 0 1 0 D F 0 02 0 D00 0 0 0 0 0 0 0 0              T=@00D00000&T;  RESET DISPLAY 'C' PENDING AND CLEAR 'ID'  5866.000
0AB8 0 0 0 3 0 A 6 0 02 0 200 0 0 0 0 0 0 0 0              DI=@00000020:T;  SET 'ID' FOR 'C'                        5867.000
0AB9 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0              S=SCRATCH(@88); FETCH 'C'                                5868.000
                                                S.P.INSERT.CDTB                                                     5869.000
0ABA 0 5 7 0 1 E 6 0 00 0 E7E 0 0 0 0 0 0 0 0  A7E        R(PNL.WRK)=S, *LINK S.TO.T.DUD;                          5870.000
0ABB 0 0 0 0 5 0 F 0 00 0 E00 0 0 0 0 0 0 0 0              T=R(PNL.WRK,HWS);                                        5871.000
0ABC 0 0 0 1 0 0 4 0 02 A 870 0 0 0 0 0 0 0 0              SCRATCH(@87)=T; PLACES DISPLAY 'C' OR 'D' IN TRANS. BUFFER  5872.000
0ABD 0 5 0 0 1 0 0 0 00 0 0A8 0 0 0 0 0 0 0 0  AA8        *GO TO S.P.TRNMT.COUNT.ADDR+2;                           5873.000
                                                S.P.RESET.DISPD.PNDG                                                5874.000
0ABE 0 5 7 0 1 0 0 0 00 0 07D 0 0 0 0 0 0 0 0  A7D        *LINK S.P.FETCH.TC.FLG.ID;                               5875.000
0ABF 0 0 0 1 0 D F 0 02 0 E00 0 0 0 0 0 0 0 0              T=@00E00000&T;  RESET DISPLAY 'D' PENDING AND CLEAR 'ID'  5876.000
0AC0 0 0 0 3 0 A 6 0 02 0 300 0 0 0 0 0 0 0 0              DI=@00000030:T;  SET 'ID' FOR 'D'                        5877.000
0AC1 0 0 0 1 1 0 1 0 02 A 8A0 0 0 0 0 0 0 0 0              S=SCRATCH(@8A);                                          5878.000
0AC2 0 5 0 0 1 0 0 0 00 0 0BA 0 0 0 0 0 0 0 0  ABA        *GO TO S.P.INSERT.CDTB;                                  5879.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    WRT.INC
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                    5880.000
                                                        S.P.WRT.INC                                 5881.000
0AC3 0 5 7 0 1 0 0 0 00 0 0E7 0 0 0 0 0 0 0 0  AE7          *LINK S.P.MEMORY.ADDRESS.TEST;             5882.000
0AC4 0 5 7 0 1 0 0 0 00 0 079 0 0 0 0 0 0 0 0  A79          *LINK S.P.FETCH.B;  GO GET B DISPLAY       5883.000
0AC5 0 4 7 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  ACC          *LINK S.P.MEMORY.WRITE; STORE B DISPLAY IN MEMORY  5884.000
0AC6 0 4 7 0 1 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  ACE          *LINK S.P.MEMORY.READ; GO VERIFY DATA WRITTEN  5885.000
0AC7 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D          IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  5886.000
0AC8 0 6 7 0 3 5 0 0 00 0 B53 0 0 0 0 0 0 0 0  B53          NOD=S-DI, *LINK S.P.DIID.CYCLE; IS MEM.DATA='B' DISPLAY DATA?  5887.000
0AC9 C 6 0 0 1 0 0 0 00 0 C4B 0 0 0 0 0 0 0 0  C4B          IF NALUZ *GO TO S.P.ERROR.6;               5888.000
0ACA 0 5 7 0 1 0 0 0 00 0 0DE 0 0 0 0 0 0 0 0  ADE          *LINK S.P.INC.A;                           5889.000
0ACB 0 5 0 0 1 0 0 0 00 0 058 0 0 0 0 0 0 0 0  A58          *GO TO S.P.INC.PC;  GO UPDATE PC TO MATCH 'A' DISPLAY  5890.000
                                                        S.P.MEMORY.WRITE                            5891.000
0ACC 0 0 0 0 1 0 0 0 1E 1 083 0 0 0 0 0 0 0 0               WRITE, FRCWORD, IGNSTOP;                   5892.000
0ACD 0 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  ACF          *GO TO S.P.RIBUSY;                         5893.000
                                                        S.P.MEMORY.READ                             5894.000
0ACE 0 0 0 0 1 0 0 0 1C 1 083 0 0 0 0 0 0 0 0               READ,FRCWORD,IGNSTOP;                      5895.000
                                                        S.P.RIBUSY                                  5896.000
0ACF 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0               *NOP;    ALLOW TIME FOR BUSY TRANSFER      5897.000
0AD0 1 4 0 0 1 0 0 0 00 0 BD0 0 0 0 0 0 0 0 0  AD0          IF RIBUSY *HOP $;  WAIT FOR MEMORY DRT     5898.000
0AD1 9 1 0 3 1 E F 0 02 0 F40 0 0 0 0 0 0 0 0               T=@000000F4, IF %SERIAL.PANEL, *JUMPJ; EXIT IF NOT SERIAL PANEL  5899.000
                                                        **       SET BMUX00 FOR NO ERROR RETURN     5900.000
0AD2 0 6 0 0 0 0 0 0 00 0 F50 0 0 0 0 0 0 0 0  F50          NOD=T, *GO TO S.P.CHECK.MAP.ERROR;         5901.000
                                                        S.P.MEMERR                                  5902.000
0AD3 9 4 0 0 3 0 0 0 00 0 0D5 0 0 0 0 0 0 0 0  AD5          NOD=DI, IF %OPTIMEOUT, *HOP $+2;           5903.000
0AD4 0 4 0 3 1 E F 0 02 0 038 0 0 0 0 0 0 0 0  AD8          T=@00000003, *HOP $+4;    NO RESPONSE FROM MEMORY  5904.000
0AD5 9 4 0 0 1 0 0 0 00 0 7D7 0 0 0 0 0 0 0 0  AD7          IF %UPNORESP, *HOP $+2;                    5905.000
0AD6 0 4 0 3 1 E F 0 02 0 048 0 0 0 0 0 0 0 0  AD8          T=@00000004, *HOP $+2;    NON-PRESENT MEMORY  5906.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                                                    WRT.TNC
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0AD7 0 0 0 3 1 E F 0 02 0 050 0 0 0 0 0 0 0 0          T=@00000005;            PARITY ERROR IN MEMORY                    5907.000
                                                **       READ LOCATION ZERO TO CLEAR LATERR FLAG. IF THIS FAILS,     5908.000
                                                **       REPORT ERROR 7, INDICATING A MAJOR BREAK.                   5909.000
                                               S.P.CLEAR.ERROR                                                      5910.000
0AD8 0 6 7 0 0 0 E 0 15 F 2FA 0 0 0 0 0 0 0 0  2FA      FULLMAR=T(ZE),CLRTO,*LINK CLEAR.MEM.ERROR; GOTO CLEAR MEM ERRORS 5911.000
0AD9 9 4 0 0 1 0 0 0 00 0 55R 0 0 0 0 0 0 0 0  ADB      IF %OPTIMEOUT:OPNORESP:OPRNDPE, *HOP $+2;                   5912.000
0ADA 0 0 0 3 1 E F 0 02 0 070 0 0 0 0 0 0 0 0          T=@00000007;            UNABLE TO READ LOCATION ZERO              5913.000
0ADB 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0          SCRATCH(@8B)=T;         ERROR CODE INTO B-DISPLAY                 5914.000
0ADC 0 0 0 7 3 A F 0 00 0 FF0 0 0 0 0 0 0 0 0          T=@FFFFFFFF:DT;         READ DI, SETUP BMUX00 FOR ERROR           5915.000
0ADD 0 1 0 0 0 0 0 0 15 0 000 0 0 0 0 0 0 0 0          NOD=T, CLRTO,*JUMPJ;                                      5916.000
                                               S.P.INC.A                                                            5917.000
0ADE 0 5 7 0 1 0 0 0 00 0 078 0 0 0 0 0 0 0 0  A78      *LINK S.P.FETCH.A;                                        5918.000
0ADF 0 0 0 3 0 3 F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004+T; INCREMENT 'A' BY 4 (FULLWORD)               5919.000
0AE0 0 1 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0          SCRATCH(@89)=T, *JUMPJ; STORE A                           5920.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                                    I N C . R D
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    5921.000
                                                        S.P.INC.RD                                                  5922.000
0AE1 0 5 7 0 1 0 0 0 00 0 0DE 0 0 0 0 0 0 0 0  ADE          *LINK S.P.INC.A;                                       5923.000
0AE2 0 4 7 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0  AE7          *LINK S.P.MEMORY.ADDRESS.TEST; GO LOAD MAR & TEST FOR MA OR PSW  5924.000
0AE3 0 5 7 0 1 0 0 0 00 0 0CE 0 0 0 0 0 0 0 0  ACE          *LINK S.P.MEMORY.READ;  FETCH MEMORY DATA ADDRESSED BY 'A' DISP  5925.000
0AE4 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D          IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  5926.000
0AE5 0 0 0 1 3 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                SCRATCH(@88)=DI; RETURNED DATA TO 'B' DISPLAY          5927.000
0AE6 0 5 0 0 1 0 0 0 00 0 058 0 0 0 0 0 0 0 0  A58          *GO TO S.P.INC.PC;                                     5928.000
                                                        S.P.MEMORY.ADDRESS.TEST                                     5929.000
0AE7 0 0 0 1 1 0 1 0 02 A 890 0 0 0 0 0 0 0 0                S=SCRATCH(@89);  FETCH A DISPLAY                       5930.000
0AE8 0 5 7 0 1 E 6 0 00 0 07A 0 0 0 0 0 0 0 0  A7A          DI=S, *LINK S.P.FETCH.C; SAVE A IN DI AND GET C DISPLAY  5931.000
0AE9 0 0 0 3 0 D E 0 02 0 010 0 0 0 0 0 0 0 0                FULLMAR=@00000001&T; CLEAR UPPER MAR & TEST FOR MA IN C DISP  5932.000
0AEA 0 0 0 3 1 E F 0 02 0 210 0 0 0 0 0 0 0 0                T=@00000021; IF MEM.ADRS. SET MEMORY ADDRESS AND MEMORY DATA  5933.000
0AEB B 4 0 4 3 D 3 0 00 D 80D 0 0 0 0 0 0 0 0  AFD          MAR=R(PCMASK)&DI, OTHERBANK, IF ALU7 *GO TO $+2; LOAD MAR MAP  5934.000
0AEC 0 4 0 0 3 0 E 0 00 0 00E 0 0 0 0 0 0 0 0  AFE          FULLMAR=DI, *GO TO S.P.C.BYTE3; LOAD MAR FOR MA 'A' DISPLAY  5935.000
0AED 0 0 0 3 1 E F 0 02 0 420 0 0 0 0 0 0 0 0                T=@00000042; SET PSW AND INSTRUCTION FLAGS             5936.000
                                                        S.P.C.BYTE3                                                 5937.000
0AEE 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0                S=SCRATCH(@88);          THIS ROUTINE CLEARS BYTE 3    5938.000
0AEF 0 0 0 7 1 E 6 0 00 0 000 0 0 0 0 0 0 0 0                DI=@FFFFFF00;            OF THE C-DISPLAY AND OR'S IN  5939.000
0AF0 0 0 0 0 3 D 1 0 00 0 000 0 0 0 0 0 0 0 0                S=S&DI;                THE CONTENTS OF THE T-REG.      5940.000
0AF1 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0                T=S:T;                 NOTICE THAT IT CLOBBERS DI.     5941.000
0AF2 0 1 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0                SCRATCH(@88)=T, *JUMPJ;                                5942.000
                                                        S.P.CLEAR                                                   5943.000
0AF3 0 0 0 3 1 E F 0 02 6 000 0 0 0 0 0 0 0 0                T=0, SDEST;                                            5944.000
0AF4 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0                SCRATCH(@88)=T;          CLEAR STOPS, ETC              5945.000
0AF5 1 4 0 0 0 0 E 0 00 0 D97 0 0 0 0 0 0 0 0  AF7          FULLMAR=T, IF UARTTBMT *GO TO $+2;                     5946.000
0AF6 A 4 3 0 1 0 0 0 00 0 905 0 0 0 0 0 0 0 0  AF5          DFCRN, IF %NCTR7 *GO TO $-1;                           5947.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

```
S F L  32/75  CPU                                                    INC.RD
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0AF7 0 0 0 0 1 0 0 0 00 0 A05 0 0 0 0 0 0 0 0          SET(UARTDAV);                    5948.000

0AF8 0 5 0 0 0 0 C 0 00 0 0B1 0 0 0 0 0 0 0 0  AB1     NU=T, *GO TO UART.XMIT+2;        5949.000
```

```
          S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          R D . O R . W R T
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    5950.000
                                                                  (@AF9)                                            5951.000
                                                                  S.P.RD.OR.WT                                      5952.000
0AF9 0 6 7 0 1 0 0 0 00 0 B54 0 0 0 0 0 0 0 0  B54                *LINK S.P.CLEAN.D;                                5953.000
0AFA 0 0 0 0 6 D 0 0 02 0 080 0 0 0 0 0 0 0 0                     MOD=@08000000&PNLDATA;                            5954.000
0AFB 0 0 0 0 6 D F 0 02 0 070 0 0 0 0 0 0 0 0                     T=@07000000&PNLDATA;                              5955.000
0AFC B 6 0 0 1 0 0 0 00 0 B9B 0 0 0 0 0 0 0 0  B9B                IF ALUZ *GO TO S.P.GPR;                           5956.000
0AFD 0 0 0 3 1 E 1 0 02 2 B00 0 0 0 0 0 0 0 0                     INIBL, S=(S.P.JUMP.TBL1+-4&@00FF); SET JUMP TABLE BASE ADDR  5957.000
0AFE 0 6 7 2 1 E 1 0 00 0 B69 0 0 0 0 0 0 0 0  B69                S=SNIBL, *LINK DUD.B6X; SHIFT BASE ADDR LFFT & MERGE VECTOR  5958.000
0AFF 6 0 1 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                     *JUMPS;                                           5959.000
                                                                  **                                                5960.000
                                                                  **       LOCATION CRITICAL JUMP TABLE - CANNOT BE MOVED.  5961.000
                                                                  **                                                5962.000
                                                                  (@B00)                                            5963.000
                                                                  S.P.JUMP.TBL1                                     5964.000
0B00 0 5 0 0 1 0 0 0 00 0 057 0 0 0 0 0 0 0 0  B57                *GO TO S.P.MEM.ADDR;                              5965.000
0B01 0 6 0 0 1 0 0 0 00 0 C1B 0 0 0 0 0 0 0 0  C1B                *GO TO S.P.PSW;                                   5966.000
0B02 0 5 0 0 1 0 0 0 00 0 06A 0 0 0 0 0 0 0 0  B6A                *GO TO S.P.PC;                                    5967.000
0B03 0 5 0 0 1 0 0 0 00 0 080 0 0 0 0 0 0 0 0  B80                *GO TO S.P.CSW;                                   5968.000
0B04 0 5 0 0 1 0 0 0 00 0 036 0 0 0 0 0 0 0 0  B36                *GO TO S.P.MEM.DATA;                              5969.000
0B05 0 5 0 0 1 0 0 0 00 0 026 0 0 0 0 0 0 0 0  B26                *GO TO S.P.EFFECTIVE.ADDRESS;                     5970.000
0B06 0 5 0 0 1 0 0 0 00 0 019 0 0 0 0 0 0 0 0  B19                *GO TO S.P.PSD2;                                  5971.000
                                                                  S.P.CONVERT                                       5972.000
0B07 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B                IF FFRUN *GO TO S.P.FAULT.2;                      5973.000
0B08 0 6 7 0 1 0 0 0 00 0 A78 0 0 0 0 0 0 0 0  A78                *LINK S.P.FETCH.A;                                5974.000
0B09 0 0 0 0 0 0 E 0 00 0 000 0 0 0 0 0 0 0 0                     FULLMAR=T;                                        5975.000
0B0A 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0                     SET(ENAUORD);                                     5976.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU

PC	T	S	M	A	B	+	D	R	Y	X	PCH	+	B	D	M	+	B	A	D	ADDR			
																					RD.DR.WRT		
0B0B	0	5	7	0	1	0	3	0	00	0	069	0	0	0	0	0	0	0	0	B69	MAR=MAR, *LINK DUP.B6X;		5977.000
0B0C	0	0	0	0	1	0	F	0	00	6	000	0	0	0	0	0	0	0	0		T=MAR, RDMAP;		5978.000
0B0D	0	0	0	1	0	0	4	0	02	A	880	0	0	0	0	0	0	0	0		SCRATCH(@88)=T;	PHYSICAL ADDR INTO DISP-B	5979.000
0B0E	0	0	0	0	1	0	0	0	00	0	005	0	0	0	0	0	0	0	0		RESET(ENAUORD);		5980.000
0B0F	0	6	7	0	1	0	0	0	00	0	A7C	0	0	0	0	0	0	0	0	A7C	*LINK S.P.FETCH.FUNCTION;		5981.000
0B10	0	0	0	0	0	8	0	0	02	0	0A0	0	0	0	0	0	0	0	0		NOD=@0A000000!T;		5982.000
0B11	0	0	0	1	1	0	1	0	02	A	880	0	0	0	0	0	0	0	0		S=SCRATCH(@88);	DISP-C	5983.000
0B12	C	6	0	0	1	E	F	0	00	0	C3C	0	0	0	0	0	0	0	0	C3C	T=S, IF NALUZ *GO TO S.P.FAULT.3;		5984.000
0B13	0	0	0	3	0	D	0	0	02	0	070	0	0	0	0	0	0	0	0		NOD=@00000007&T;		5985.000
0B14	0	0	0	7	0	D	F	0	00	0	870	0	0	0	0	0	0	0	0		T=@FFFFFF87&T;		5986.000
0B15	B	6	0	0	1	0	0	0	00	0	C3C	0	0	0	0	0	0	0	0	C3C	IF ALUZ *GO TO S.P.FAULT.3;		5987.000
0B16	0	0	0	3	0	A	F	0	02	0	100	0	0	0	0	0	0	0	0		T=@00000010:T;	SET EFFECTIVE ADDRESS	5988.000
0B17	0	0	0	1	0	0	4	0	02	A	880	0	0	0	0	0	0	0	0		SCRATCH(@88)=T;		5989.000
0B18	0	6	0	0	1	0	0	0	00	0	C0B	0	0	0	0	0	0	0	0	C0B	*GO TO S.P.SET.B.C.D;		5990.000
																					S.P.PSD2		5991.000
0B19	3	6	0	0	1	0	0	0	00	3	C3B	0	0	0	0	0	0	0	0	C3B	IF FFRUN *GO TO S.P.FAULT.2;		5992.000
0B1A	0	6	7	0	1	0	0	0	00	0	A7C	0	0	0	0	0	0	0	0	A7C	*LINK S.P.FETCH.FUNCTION;		5993.000
0B1B	0	0	0	0	0	8	0	0	02	0	0B0	0	0	0	0	0	0	0	0		NOD=@0B000000!T;		5994.000
0B1C	0	0	0	1	1	0	1	0	02	A	900	0	0	0	0	0	0	0	0		S=SCRATCH(@90);		5995.000
0B1D	B	4	0	0	1	E	F	0	00	0	00F	0	0	0	0	0	0	0	0	B1F	T=S, IF ALUZ *GO TO $+2;		5996.000
0B1E	0	4	0	1	0	0	4	0	02	A	8B8	0	0	0	0	0	0	0	0	B18	SCRATCH(@8B)=T, *HOP $-6;	READ PSD2 EXIT	5997.000
0B1F	0	6	7	0	1	0	0	0	00	0	A7D	0	0	0	0	0	0	0	0	A7D	*LINK S.P.FETCH.TC.FLG.TO;	WRITE PSD2 START	5998.000
0B20	0	0	0	0	0	A	F	0	02	0	040	0	0	0	0	0	0	0	0		T=@04000000:T;		5999.000
0B21	0	0	0	1	0	0	4	0	02	A	860	0	0	0	0	0	0	0	0		SCRATCH(@86)=T;		6000.000
0B22	0	6	7	0	1	0	0	0	00	0	A79	0	0	0	0	0	0	0	0	A79	*LINK S.P.FETCH.B;		6001.000
0B23	0	0	0	0	1	0	0	0	00	0	004	0	0	0	0	0	0	0	0		RESET(HIREG);		6002.000
0B24	0	6	0	0	0	0	0	0	00	0	9CE	0	0	0	0	0	0	0	0	9CE	NOD=T, *GO TO PANEL.EVALUATE.PSD2;		6003.000

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          R D . O R . W R T
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    (@R26)                                                          6004.000

                                                    S.P.EFFECTIVE.ADDRESS                                           6005.000

0B26 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B        IF FFRUN *GO TO S.P.FAULT.2;                             6006.000

0B27 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C        *LINK S.P.FETCH.FUNCTION;                                6007.000

0B28 0 0 0 0 0 8 0 0 02 0 0B0 0 0 0 0 0 0 0 0              NOD=@0B000000!T;                                         6008.000

0B29 0 0 0 0 1 0 0 0 00 8 007 0 0 0 0 0 0 0 0              PCTOMAR,CLRS;                                            6009.000

0B2A B 6 0 0 1 0 0 0 00 0 C3D 0 0 0 0 0 0 0 0  C3D        IF ALUZ *GO TO S.P.FAULT.4;                              6010.000

0B2B 0 6 7 0 1 0 0 0 00 0 226 0 0 0 0 0 0 0 0  226        *LINK READ.EFFECTIVE.ADDR;   GO READ EFFECTIVE ADDR      6011.000

0B2C 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0              SCRATCH(@8B)=T;          PUT TT IN B-DISP                6012.000

0B2D 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A        *LINK S.P.FETCH.C;                                       6013.000

0B2E 0 0 0 7 0 D F 0 00 0 070 0 0 0 0 0 0 0 0              T=@FFFFFF07&T;                                           6014.000

0B2F 0 0 0 3 0 D 0 0 02 0 090 0 0 0 0 0 0 0 0              NOD=@00000009&T;         IF %PSW & %PC THEN OP FAULT     6015.000

0B30 0 0 0 3 0 A F 0 02 0 100 0 0 0 0 0 0 0 0              T=@00000010:T;           OR IN EA                        6016.000

0B31 C 6 0 0 1 0 0 0 00 0 C3D 0 0 0 0 0 0 0 0  C3D        IF NALUZ *GO TO S.P.FAULT.4;                             6017.000

0B32 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0              SCRATCH(@8B)=T;          C DISPLAY                       6018.000

0B33 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0              T=@00000000;                                             6019.000

0B34 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0              SCRATCH(@8A)=T;          D DISPLAY                       6020.000

0B35 0 6 0 0 1 0 0 0 00 0 C0B 0 0 0 0 0 0 0 0  C0B        *GO TO S.P.SET.B.C.D;                                    6021.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU MEM.DATA

```
PC   T S M A B + D R  Y X PCH + R D M + R A D  ADDR
                                                                                                                    6022.000
                                                                                                                    6023.000
                                                S.P.MEM.DATA
0836 0 0 0 1 1 0 1 0 02 A 890 0 0 0 0 0 0 0 0          S=SCRATCH(@89);  FETCH A DISPLAY                          6024.000
0837 0 6 7 0 1 E 6 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A     DI=S, *LINK S.P.FETCH.C;  SAVE A DISP IN DI & GO GET C DISP 6025.000
0838 0 0 0 3 0 D E 0 02 0 010 0 0 0 0 0 0 0 0          FULLMAR=@00000001&T; CLEAR UPPER MAR & TEST FOR MA IN 'C'  6026.000
0839 0 0 0 1 1 0 1 0 02 A 8C0 0 0 0 0 0 0 0 0          S=SCRATCH(@8C); FETCH FUNCTION CODE                       6027.000
083A C 4 0 0 1 E F 0 00 0 00C 0 0 0 0 0 0 0 0  B3C     T=S, IF NALUZ *GO TO S.P.MA.IN.A; BRANCH IF MA IN 'C'      6028.000
083B 0 4 0 4 3 D 3 0 00 D B0D 0 0 0 0 0 0 0 0  B3D     MAR=R(PCMASK)&DI, OTHERBANK, *GO TO S.P.CHECK.FUNCTION; LMAR MAP 6029.000
                                                S.P.MA.IN.A                                                         6030.000
083C 0 0 0 0 3 0 E 0 00 0 000 0 0 0 0 0 0 0 0          FULLMAR=DI;  LOAD MAR UNMAPPED                            6031.000
```

```
                                                        S.P.CHECK.FUNCTION                                                  6032.000
0B3D 0 0 0 0 0 3 0 0 02 0 060 0 0 0 0 0 0 0 0                   NOD=@06000000+T; TEST FOR READ FUNCTION CODE                6033.000
0B3E 0 0 0 1 1 0 1 0 02 A 8B0 0 0 0 0 0 0 0 0                   S=SCRATCH(@8B); FETCH THE B DISPLAY FOR A MEM WRITE         6034.000
0B3F 2 4 0 0 1 0 0 0 00 0 706 0 0 0 0 0 0 0 0  B46              IF ALU4-7Z *GO TO S.P.READ;                                 6035.000
                                                        S.P.WRITE                                                           6036.000
0B40 0 6 7 0 1 E F 0 00 0 ACC 0 0 0 0 0 0 0 0  ACC              T=S, *LINK S.P.MEMORY.WRITE; MOVE 'B' TO T REG & GO WRITE TO MEM 6037.000
0B41 3 6 7 0 0 0 0 0 00 4 ACE 0 0 0 0 0 0 0 0  ACE              NOD=T, IF BMUX00 *LINK S.P.MEMORY.READ; DON'T READ IF WRITE ERR 6038.000
0B42 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D              IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.   6039.000
0B43 0 5 7 0 3 5 0 0 00 0 069 0 0 0 0 0 0 0 0  B69              NOD=S-DI, *LINK DUD.B6X; IS MEM DATA = B DISPLAY DATA?      6040.000
0B44 B 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  B49              IF ALUZ *GO TO S.P.MA;                                      6041.000
0B45 0 6 0 0 1 0 0 0 00 0 C4B 0 0 0 0 0 0 0 0  C4B              *GO TO S.P.ERROR.6;                                         6042.000
                                                        S.P.READ                                                            6043.000
0B46 0 6 7 0 1 0 0 0 00 0 ACE 0 0 0 0 0 0 0 0  ACE              *LINK S.P.MEMORY.READ;                                      6044.000
0B47 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D              IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.   6045.000
0B48 0 0 0 1 3 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                   SCRATCH(@8B)=DI; MOVE DI TO 'B' DISPLAY                     6046.000
                                                        S.P.MA                                                              6047.000
0B49 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A              *LINK S.P.FETCH.C;                                          6048.000
0B4A 0 0 0 7 0 D 6 0 00 0 0F0 0 0 0 0 0 0 0 0                   DI=@FFFFFF0F&T;          SAVE ALL BUT MA, PSW, ETC          6049.000
0B4B 0 0 0 7 0 A F 0 00 0 FE0 0 0 0 0 0 0 0 0                   T=@FFFFFFFE:T;           TEST FOR MA                        6050.000
0B4C 0 0 0 3 0 3 0 0 02 0 010 0 0 0 0 0 0 0 0                   NOD=1+T;                                                    6051.000
0B4D 0 0 0 3 1 E F 0 02 0 210 0 0 0 0 0 0 0 0                   T=@00000021;            ASSUME MA                          6052.000
0B4E B 5 0 0 1 0 0 0 00 0 050 0 0 0 0 0 0 0 0  B50              IF ALUZ *GO TO $+2;       IF MA, USE 21                    6053.000
0B4F 0 0 0 3 3 A F 0 02 0 400 0 0 0 0 0 0 0 0                   T=@00000040:DI;         NOT MA, USE INST & DI              6054.000
0B50 0 6 7 0 1 0 0 0 00 0 AEE 0 0 0 0 0 0 0 0  AEE              *LINK S.P.C.BYTE3;                                          6055.000
0B51 0 4 7 1 1 E F 0 02 0 705 0 0 0 0 0 0 0 0  B55              T=@00700000, *LINK S.P.CLEAN.D+1;   SET B,C,D, CLEAR D      6056.000
0B52 0 6 0 0 1 0 0 0 00 0 C0D 0 0 0 0 0 0 0 0  C0D              *GO TO S.P.SAVE.FLAG;                                       6057.000
                                                        S.P.DUD.CYCLE                                                       6058.000
0B53 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   *JUMPJ;                                                     6059.000
```

```
                                                    S.P.CLEAN.D                                                    6060.000
0B54 0 6 7 0 1 0 0 0 00 0 A7B 0 0 0 0 0 0 0 0  A7B      *LINK S.P.FETCH.D;                                        6061.000
0B55 0 0 0 7 0 D F 0 00 0 FC0 0 0 0 0 0 0 0 0           T=@FFFFFFFC&T;                                            6062.000
0B56 0 1 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0           SCRATCH(@8A)=T, *JUMPJ;                                   6063.000
                                                    S.P.MEM.ADDR                                                   6064.000
0B57 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C      *LINK S.P.FETCH.FUNCTION;                                 6065.000
0B58 0 0 0 0 0 8 0 0 02 0 0B0 0 0 0 0 0 0 0 0           NOD=@0B000000!T;        TEST FOT WRITE                    6066.000
0B59 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                     6067.000
0B5A C 6 0 0 1 0 0 0 00 0 C3C 0 0 0 0 0 0 0 0  C3C      IF NALUZ *GO TO S.P.FAULT.3;                              6068.000
0B5B 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79      *LINK S.P.FETCH.B;                                        6069.000
0B5C 0 0 0 7 0 D F 0 00 0 FC0 0 0 0 0 0 0 0 0           T=@FFFFFFFC&T;  FORCE FULLWORD                            6070.000
0B5D 0 0 0 4 0 D F 0 02 0 000 0 0 0 0 0 0 0 0           T=@00FFFFFF&T; MASK OUT BITS 0-7                          6071.000
0B5E 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0           SCRATCH(@89)=T; STORE A                                   6072.000
0B5F 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A      *LINK S.P.FETCH.C;                                        6073.000
0B60 0 0 0 3 0 D F 0 02 0 F00 0 0 0 0 0 0 0 0           T=@000000F0&T;                                            6074.000
0B61 0 0 0 7 0 3 0 0 00 0 C00 0 0 0 0 0 0 0 0           NOD=@FFFFFFC0+T; TEST FOR INSTRUCTION                     6075.000
0B62 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0           T=S;                                                      6076.000
0B63 B 4 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0  B66      IF ALUZ *GO TO $+3;                                       6077.000
0B64 0 0 0 2 0 D F 0 02 0 0F0 0 0 0 0 0 0 0 0           T=@00000F00&T; IF %INSTRUCTION IT IS KEYBOARD, LET KB PASS BY  6078.000
0B65 0 0 0 3 1 E F 0 02 0 010 0 0 0 0 0 0 0 0           T=@00000001;                    SET MEMORY ADDRESS        6079.000
0B66 0 6 7 0 1 0 0 0 00 0 AEE 0 0 0 0 0 0 0 0  AEE      *LINK S.P.C.BYTE3;                                        6080.000
                                                    S.P.FLAG.ADDR                                                  6081.000
0B67 0 5 7 0 1 0 0 0 00 0 054 0 0 0 0 0 0 0 0  B54      *LINK S.P.CLEAN.D;                                        6082.000
0B68 0 6 0 0 1 0 0 0 00 0 C0A 0 0 0 0 0 0 0 0  C0A      *GO TO S.P.SET.A.C.D;             RESOLUTION              6083.000
                                                    DUD.B6X                                                        6084.000
0B69 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPJ;                                                   6085.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    PROGRAM COUNTER
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6086.000
                                                        S.P.PC                                                      6087.000
0B6A 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B          IF FFRUN *GO TO S.P.FAULT.2;                            6088.000
0B6B 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C          *LINK S.P.FETCH.FUNCTION;                               6089.000
0B6C 0 0 0 0 0 8 0 0 02 8 0A0 0 0 0 0 0 0 0 0               NOD=@0A000000!T, PCTOMAR; TEST FOR READ                 6090.000
0B6D 6 0 0 6 4 D 6 7 02 D 7F0 0 0 0 0 0 0 0 0               DI=@FFFF7FFF&FR(TRACE), OTHERBANK; CLEAR RHFLAG          6091.000
0B6E B 5 0 4 1 D F 0 00 D 875 0 0 0 0 0 0 0 0  B75          T=R(PCMASK)&MAR, OTHERBANK, IF ALUZ *GO TO S.P.READ.PC; 6092.000
                                                        S.P.WRITE.PC                                                6093.000
0B6F 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79          *LINK S.P.FETCH.B;  GO GET B DISPLAY (NEW PC VALUE)      6094.000
0B70 0 0 0 3 0 D 0 0 02 0 020 0 0 0 0 0 0 0 0               NOD=@00000002&T;  TEST FOR RHFLAG IN NEW PC             6095.000
0B71 0 0 0 4 0 D 2 0 00 D 800 0 0 0 0 0 0 0 0               PC=R(PCMASK)&T, OTHERBANK, CLDNU; SET NEW PC VALUE      6096.000
0B72 B 4 0 0 3 0 8 0 00 8 F04 0 0 0 0 0 0 0 0  B74          R(TRACE)=DI, PCTOMAR, IF ALUZ *GO TO $+2; BRANCH IF NO RHFLAG 6097.000
0B73 6 0 0 2 0 A B 7 02 D 800 0 0 0 0 0 0 0 0               FR(TRACE)=@00008000:T, OTHERBANK; SET RHFLAG IN R(TRACE) 6098.000
0B74 0 0 0 0 1 0 F 0 00 D 000 0 0 0 0 0 0 0 0               T=MAR, OTHERBANK;  GET NEW PC VALUE IN T                6099.000
                                                        S.P.READ.PC                                                 6100.000
0B75 0 0 0 0 4 0 0 0 00 D F00 0 0 0 0 0 0 0 0               NOD=R(TRACE), OTHERBANK;  TEST FOR RHFLAG IN R(TRACE)    6101.000
0B76 3 4 0 0 1 0 0 0 00 8 008 0 0 0 0 0 0 0 0  B78          IF BMUX16 *GO TO $+2;                                   6102.000
0B77 0 0 0 3 0 A F 0 02 0 020 0 0 0 0 0 0 0 0               T=@00000002:T; SET RHFLAG IN PC VALUE                   6103.000
0B78 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0               SCRATCH(@89)=T; SAVE A DISPLAY (PC VALUE)               6104.000
0B79 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A          *LINK S.P.FETCH.C;                                      6105.000
0B7A 0 0 0 3 0 D F 0 02 0 400 0 0 0 0 0 0 0 0               T=@00000040&T;                                          6106.000
0B7B 0 0 0 3 1 E F 0 02 0 440 0 0 0 0 0 0 0 0               T=@00000044 ; IF INSTR. SET INSTR. AND PROGRAM COUNTER  6107.000
0B7C C 4 0 0 1 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  B7E          IF NALUZ *GO TO $+2;                                    6108.000
0B7D 0 0 0 3 1 E F 0 02 0 040 0 0 0 0 0 0 0 0               T=@00000004; IF NOT INSTRUCTION, SET PROGRAM COUNTER    6109.000
0B7E 0 6 7 0 1 0 0 0 00 0 AEE 0 0 0 0 0 0 0 0  AEE          *LINK S.P.C.BYTE3;                                      6110.000
0B7F 0 5 0 0 1 0 0 0 00 0 067 0 0 0 0 0 0 0 0  B67          *GO TO S.P.FLAG.ADDR;                                   6111.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU CONTROL SWITCHES

```
PC   T S M A B + D B  Y X PCH + B D M + B A D  ADDR

                                                                                                                6112.000
                                                                    S.P.CSW                                     6113.000
0B80 0 0 0 3 1 E F 0 02 6 78D 0 0 0 0 0 0 0 0                           T=@00000078, SDEST, CLDNU;                  6114.000
0B81 0 6 7 2 1 E E 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C                      FULLMAR=SN1BL, *LINK S.P.FETCH.FUNCTION; (S=@00000780)  6115.000
0B82 0 0 0 0 0 8 0 0 02 0 0A0 0 0 0 0 0 0 0 0                           NOD=@0A000000!T; TEST FUNCTION FOR A READ   6116.000
0B83 0 0 0 7 1 3 0 0 00 0 FC0 0 0 0 0 0 0 0 0                           NOD=@FFFFFFFC+MAR;                          6117.000
0B84 C 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  B89                      IF NALUZ *GO TO S.P.CSW.WRITE; IS FUNCTION A READ OR A WRITE ?  6118.000
                                                                    S.P.CSW.READ                                6119.000
0B85 0 6 7 0 1 0 0 0 00 0 ACE 0 0 0 0 0 0 0 0  ACE                      *LINK S.P.MEMORY.READ;                      6120.000
0B86 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D                      IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  6121.000
0B87 0 0 0 1 3 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                           SCRATCH(@8B)=DI;                            6122.000
0B88 0 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  B8F                      *GO TO S.P.CSW.END;                         6123.000
                                                                    S.P.CSW.WRITE                               6124.000
0B89 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79                      *LINK S.P.FETCH.B; FETCH DATA TO BE WRITTEN IN MEMORY  6125.000
0B8A 0 6 7 0 1 0 0 0 00 0 ACC 0 0 0 0 0 0 0 0  ACC                      *LINK S.P.MEMORY.WRITE;  GO STORE B DISPLAY IN MEMORY  6126.000
0B8B 0 6 7 0 1 0 0 0 00 0 ACE 0 0 0 0 0 0 0 0  ACE                      *LINK S.P.MEMORY.READ; GO VERIFY DATA WRITTEN  6127.000
0B8C 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D                      IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  6128.000
0B8D 0 5 7 0 3 5 F 0 00 0 069 0 0 0 0 0 0 0 0  B69                      T=S-DI, *LINK DUD.B6X; COMAPRE MEMORY DATA TO B DISPLAY DATA  6129.000
0B8E C 6 0 0 1 0 0 0 00 0 C4B 0 0 0 0 0 0 0 0  C4B                      IF NALUZ *GO TO S.P.ERROR.6; COMPARE ERROR FOUND ?  6130.000
                                                                    S.P.CSW.END                                 6131.000
0B8F 0 0 0 1 1 0 4 0 02 A 890 0 0 0 0 0 0 0 0                           SCRATCH(@89)=MAR;                           6132.000
0B90 0 0 0 3 1 E F 0 02 0 010 0 0 0 0 0 0 0 0                           T=@00000001;            SET 'MA'            6133.000
0B91 0 6 7 0 1 0 0 0 00 0 AEE 0 0 0 0 0 0 0 0  AEE                      *LINK S.P.C.BYTE3;        ...IN C-DISPLAY   6134.000
0B92 0 0 0 3 1 E F 0 02 0 020 0 0 0 0 0 0 0 0                           T=@00000002; SET CSW INDICATOR              6135.000
0B93 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0                           SCRATCH(@8A)=T; STORE IN 'D' DISPLAY        6136.000
0B94 0 6 0 0 1 0 0 0 00 0 C0C 0 0 0 0 0 0 0 0  C0C                      *GO TO S.P.SET.ALL.FLAGS;                   6137.000
                                                                    S.P.KBDATA                                  6138.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          C O N T R O L  S W I T C H E S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0895 0 0 0 0 6 D 1 0 02 0 0F0 0 0 0 0 0 0 0 0          S=@0F000000&PNLDATA;  CLEAR UNUSED BITS IN PNLDATA                6139.000

0896 0 6 7 2 1 E 6 0 00 0 A79 0 0 0 0 0 0 0 0  A79    DI=SNIBL *LINK S.P.FETCH.B; PLACE DATA IN MSN OF T REG             6140.000

0897 0 0 0 0 3 0 F 0 00 0 000 0 0 0 0 0 0 0 0          T=DI;                                                             6141.000

0898 0 0 0 2 1 E F 0 00 2 000 0 0 0 0 0 0 0 0          T=SNIBL,TNIBL;        SHIFT IN NEW DIGIT                          6142.000

0899 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0          SCRATCH(@8B)=T;       WRITE 'B' TO SCRATCH PAD                    6143.000

089A 0 6 0 0 1 0 0 0 00 0 C09 0 0 0 0 0 0 0 0  C09    *GO TO S.P.SET.B;                                                 6144.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU G.P.REGISTERS

```
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR

                                                                                                    6145.000
                                                    S.P.GPR                                         6146.000
0B98 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B      IF FFRUN *GO TO S.P.FAULT.2;                6147.000
0B9C 0 0 0 0 6 D F 0 02 0 0F0 0 0 0 0 0 0 0 0           T=@0F000000&PNLDATA;                        6148.000
0B9D 0 0 0 0 0 0 E 0 00 0 00D 0 0 0 0 0 0 0 0           FULLMAR=T, CLDNU;    LOAD IN NCTR0-7        6149.000
0B9E 0 6 7 0 1 1 C 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C      NU=%N *LINK S.P.FETCH.FUNCTION;             6150.000
0B9F 0 0 0 0 0 3 0 0 02 0 060 0 0 0 0 0 0 0 0           NOD=@06000000+T;        TEST FOR READ       6151.000
0BA0 0 0 0 1 1 0 1 0 02 A 8B0 0 0 0 0 0 0 0 0           S=SCRATCH(@8B); FETCH 'B'                   6152.000
0BA1 2 4 0 0 1 0 0 0 00 0 703 0 0 0 0 0 0 0 0  BA3      IF ALU4-7Z *GO TO $+2;                      6153.000
                                                    S.P.GPR.WRITE                                   6154.000
0BA2 0 4 0 0 1 E B 3 00 0 004 0 0 0 0 0 0 0 0  BA4      R(NCTR)=S *GO TO $+2;                       6155.000
                                                    S.P.GPR.READ                                    6156.000
0BA3 0 0 0 0 4 0 1 3 00 D 000 0 0 0 0 0 0 0 0           S=R(NCTR), OTHERBANK;                       6157.000
0BA4 0 0 0 0 1 C 6 0 02 D FF0 0 0 0 0 0 0 0 0           DI=@FF000000&%N, OTHERBANK; STORE THE REGISTER NUMBER IN 'DI'  6158.000
0BA5 0 0 0 0 1 C F 0 02 0 010 0 0 0 0 0 0 0 0           T=@01000000&%N; MASK BIT DEFINING ODD OR EVEN REG GROUP  6159.000
0BA6 0 0 0 0 0 8 0 0 02 0 010 0 0 0 0 0 0 0 0           NOD=@01000000!T; TEST FOR ODD REGISTERS     6160.000
0BA7 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0           T=S;                                        6161.000
0BA8 B 5 0 0 1 0 0 0 00 0 0B4 0 0 0 0 0 0 0 0  BB4      IF ALUZ *GO TO S.P.ODDREG; ODD REGISTERS DETECTED  6162.000
                                                    S.P.EVENREG                                     6163.000
0BA9 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0           SCRATCH(@89)=T; STORE A                     6164.000
0BAA 0 0 0 0 3 0 8 0 00 0 E00 0 0 0 0 0 0 0 0           R(PNL.WRK)=DI;                              6165.000
0BAB 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A      *LINK S.P.FETCH.C;        TO CLEAR 'C' APPROPRIATELY  6166.000
0BAC 0 0 0 7 0 D F 0 00 0 F80 0 0 0 0 0 0 0 0           T=@FFFFFFF8&T;                              6167.000
0BAD 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0           SCRATCH(@88)=T;                             6168.000
0BAE 0 6 7 0 1 0 0 0 00 0 A7B 0 0 0 0 0 0 0 0  A7B      *LINK S.P.FETCH.D;        TO AVOID CLEARING OPP REG DISP  6169.000
0BAF 0 0 0 6 0 D 1 0 02 0 F00 0 0 0 0 0 0 0 0           S=@FFFFF0FF&T;                              6170.000
0BB0 0 0 0 0 5 A F 0 00 0 E00 0 0 0 0 0 0 0 0           T=S:R(PNL.WRK,HWS);                         6171.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU G.P.REGISTERS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
08B1 0 0 0 3 0 A F 0 02 0 080 0 0 0 0 0 0 0 0          T=@00000008:T;        SET DISPLAY 'D' WITH EVEN REG.          6172.000
08B2 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0          SCRATCH(@8A)=T; STORE D                               6173.000
08B3 0 6 0 0 1 0 0 0 00 0 C0A 0 0 0 0 0 0 0 0 C0A      *GO TO S.P.SET.A.C.D;                                 6174.000
                                                  S.P.ODDREG                                                 6175.000
08B4 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0          SCRATCH(@8B)=T; STORE 'B'                             6176.000
08B5 0 0 0 0 3 0 8 0 00 0 E00 0 0 0 0 0 0 0 0          R(PNL.WRK)=DI;                                        6177.000
08B6 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0 A7A      *LINK S.P.FETCH.C;       AS ABOVE FOR ODD REG          6178.000
08B7 0 0 0 7 0 D F 0 00 0 0F0 0 0 0 0 0 0 0 0          T=@FFFFFF0F&T;                                        6179.000
08B8 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0          SCRATCH(@88)=T;                                       6180.000
08B9 0 0 0 1 1 0 1 0 02 A 8A0 0 0 0 0 0 0 0 0          S=SCRATCH(@8A);                                       6181.000
08BA 0 0 0 2 1 E F 0 00 0 000 0 0 0 0 0 0 0 0          T=SNIBL;                                              6182.000
08BB 0 0 0 6 0 D 6 0 02 0 F00 0 0 0 0 0 0 0 0          DI=@FFFFF0FF&T;                                       6183.000
08BC 0 0 0 7 3 D 1 0 00 0 FD0 0 0 0 0 0 0 0 0          S=@FFFFFFFD&DI;         TURN OFF CSWS                 6184.000
08BD 0 0 0 0 5 A 1 0 00 0 E00 0 0 0 0 0 0 0 0          S=S:R(PNL.WRK,HWS);                                   6185.000
08BE 0 0 0 3 1 E F 0 00 0 000 0 0 0 0 0 0 0 0          T=SNIBR;             PLACE ODD REG.NO. IN 24-27       6186.000
08BF 0 0 0 3 0 A F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004:T;       SET DISPLAY 'D' WITH ODD  REG. ENABLE 6187.000
08C0 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0          SCRATCH(@8A)=T; STORE D                               6188.000
08C1 0 6 0 0 1 0 0 0 00 0 C0B 0 0 0 0 0 0 0 0 C0B      *GO TO S.P.SET.B.C.D;                                 6189.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          F L G . D I S P C . P N D G
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR

                                                                                                                  6190.000

                                                S.P.KEYBOARD                                                      6191.000

0BC2 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C        *LINK S.P.FETCH.FUNCTION;                              6192.000

0BC3 0 0 0 0 0 8 0 0 02 0 080 0 0 0 0 0 0 0 0             NOU=@08000000!T;        TEST FOR KEYBOARD FUNCTION    6193.000

0BC4 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0             T=@00000000;                                           6194.000

0BC5 C 4 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  BCD        IF NALU7 *GO TO S.P.EXTENDED.1;                        6195.000

0BC6 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0             SCRATCH(@8A)=T; CLEAR THE D DISPLAY                    6196.000

0BC7 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0             SCRATCH(@8B)=T; STORE B                                6197.000

0BC8 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A        *LINK S.P.FETCH.C;                                     6198.000

0BC9 0 0 0 7 0 D F 0 00 0 070 0 0 0 0 0 0 0 0             T=@FFFFFF07&T;          CLEAR OP-FLT,MD,EA,INST,STOP  6199.000

0BCA 0 0 0 2 0 A F 0 02 0 080 0 0 0 0 0 0 0 0             T=@00000800:T;                SET KEYBOARD             6200.000

0BCB 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0             SCRATCH(@88)=T; STORE C                                6201.000

0BCC 0 6 0 0 1 0 0 0 00 0 C0B 0 0 0 0 0 0 0 0  C0B        *GO TO S.P.SET.B.C.D;                                  6202.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          E X T E N D E D . 1
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6203.000

                                                        S.P.EXTENDED.1                                              6204.000

08CD 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C          *LINK S.P.FETCH.FUNCTION;                                  6205.000

08CE 0 0 0 0 0 8 0 0 02 0 0C0 0 0 0 0 0 0 0 0                NOD=@0C000000!T;                                           6206.000

08CF 0 0 0 1 1 0 1 0 02 A 890 0 0 0 0 0 0 0 0                S=SCRATCH(@89); FFTCH 'A' DISPLAY                          6207.000

08D0 C 6 0 0 1 0 0 0 00 0 C0B 0 0 0 0 0 0 0 0  C0B          IF NALUZ *GO TO S.P.SET.B.C.D;                             6208.000

08D1 0 6 7 0 1 E E 0 00 0 ACE 0 0 0 0 0 0 0 0  ACE          FULLMAR=S, *LINK S.P.MEMORY.READ;                          6209.000

08D2 5 6 0 0 1 0 0 0 00 4 C4D 0 0 0 0 0 0 0 0  C4D          IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  6210.000

08D3 0 0 0 1 3 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                SCRATCH(@8B)=DI; STORE DATA READ IN 'B' DISPLAY             6211.000

08D4 0 6 0 0 1 0 0 0 00 0 C09 0 0 0 0 0 0 0 0  C09          *GO TO S.P.SET.B;                                          6212.000

                                                        S.P.INST.STEP                                               6213.000

08D5 3 6 0 0 1 0 0 0 00 3 C3B 0 0 0 0 0 0 0 0  C3B          IF FFRUN *GO TO S.P.FAULT.2;                               6214.000

08D6 0 0 0 0 0 1 0 0 0 00 0 205 0 0 0 0 0 0 0 0              RESET(UARTDAV);                                            6215.000

08D7 0 6 0 0 1 0 0 0 00 0 218 0 0 0 0 0 0 0 0  218          *GO TO S.P.INSTR.STEP;                                     6216.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU INSTRUCTION STOP

```
PC   T S M A B + D R  Y X PCH + B D M + R A D  ADDR
                                                                                                                    6217.000
                                                                S.P.SET.C.FLAG                                      6218.000
08D8 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0                       S=SCRATCH(@88);                                 6219.000
08D9 0 0 0 0 0 A F 0 00 0 000 0 0 0 0 0 0 0 0                       T=S:T;                                          6220.000
08DA 0 0 0 1 0 0 4 0 02 A 860 0 0 0 0 0 0 0 0                       SCRATCH(@88)=T;                                 6221.000
08DB 0 0 0 1 1 E F 0 02 0 300 0 0 0 0 0 0 0 0                       T=@00300000; SET FLAG FOR C                     6222.000
08DC 0 0 0 1 0 0 4 0 02 A 860 0 0 0 0 0 0 0 0                       SCRATCH(@86)=T; STORE FLAG BIT                  6223.000
08DD 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79                  *LINK S.P.FETCH.B;                              6224.000
08DE 0 1 0 7 0 0 F 0 00 0 FC0 0 0 0 0 0 0 0 0                       T=@FFFFFFFC&T,*JUMPJ;  CLEAR C BITS FROM STOP ADDRESS  6225.000
                                                                S.P.INST.STOP                                       6226.000
08DF 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A                  *LINK S.P.FETCH.C;                              6227.000
08E0 0 0 0 2 0 D 0 0 02 0 010 0 0 0 0 0 0 0 0                       NOD=@00000100&T; TEST FOR INST.STOP             6228.000
08E1 0 0 0 2 1 E F 0 02 0 010 0 0 0 0 0 0 0 0                       T=@00000100; IF NOT OFF TURN ON                 6229.000
08E2 C 5 0 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0  BF0                  IF NALUZ *GO TO TURN.OFF.STOP;                  6230.000
08E3 0 5 7 0 1 0 0 0 00 0 0D8 0 0 0 0 0 0 0 0  BD8                  *LINK S.P.SET.C.FLAG;                           6231.000
08E4 0 4 0 0 0 A F 0 02 0 058 0 0 0 0 0 0 0 0  BE8                  T=@05000000:T, *HOP S.P.WRT.RD.STOP; INST STOP CODE  6232.000
```

```
                SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                  OPRND.RD.STOP
PC   TSMAB+DR  YX PCH+BDM+BAD  ADDR
                                                                                6233.000
                                                S.P.OPRND.RD.STOP               6234.000
0BE5 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A       *LINK S.P.FETCH.C;    6235.000
0BE6 0 0 0 2 0 D 0 0 02 0 020 0 0 0 0 0 0 0 0            NOD=@00000200&T; TEST FOR OPERAND READ STOP  6236.000
0BE7 0 0 0 2 1 E F 0 02 0 020 0 0 0 0 0 0 0 0            T=@00000200; IF NOT OFF TURN OFF  6237.000
0BE8 C 5 0 0 1 0 0 0 00 0 0F0 0 0 0 0 0 0 0 0  BF0       IF NALUZ *GO TO TURN.OFF.STOP;  6238.000
0BE9 0 5 7 0 1 0 0 0 00 0 0D8 0 0 0 0 0 0 0 0  BD8       *LINK S.P.SET.C.FLAG;  6239.000
0BEA 0 0 0 0 0 A F 0 02 0 010 0 0 0 0 0 0 0 0            T=@01000000:T; SET OP.RD  6240.000
                                                S.P.WRT.RD.STOP                 6241.000
0BEB 0 0 0 0 0 1 1 0 00 0 000 0 0 0 0 0 0 0 0            S=%T; PUTS COMPLEMENTED DATA ON D-BUS  6242.000
0BEC 0 0 0 0 0 1 1 0 00 0 805 0 0 0 0 0 0 0 0            S=%T,SET(ENAUORD); SEND DATA TO D-BUS  6243.000
0BED 0 0 0 0 0 1 1 0 00 9 000 0 0 0 0 0 0 0 0            S=%T,LDSTOP; LOAD COMPARE REGISTERS  6244.000
0BEE 0 0 0 0 0 1 1 0 00 0 005 0 0 0 0 0 0 0 0            S=%T,RESET(ENAUORD);  6245.000
0BEF 0 6 0 0 1 0 0 0 00 0 A8D 0 0 0 0 0 0 0 0  A8D       *GO TO S.P.UART.TRMT;  6246.000
                                                TURN.OFF.STOP                   6247.000
0BF0 0 0 0 1 1 E 6 0 02 0 300 0 0 0 0 0 0 0 0            DI=@00300000; SET FLAG FOR 'C'  6248.000
0BF1 0 0 0 1 3 0 4 0 02 A 860 0 0 0 0 0 0 0 0            SCRATCH(@86)=DI; SAVE FLAG BIT  6249.000
0BF2 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0            S=SCRATCH(@88);  6250.000
0BF3 0 0 0 3 0 A F 0 02 0 800 0 0 0 0 0 0 0 0            T=@00000080:T;          TURN OFF STOP LITE TOO  6251.000
0BF4 0 0 0 0 0 C F 0 00 0 000 0 0 0 0 0 0 0 0            T=S&%T;  6252.000
0BF5 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0            SCRATCH(@88)=T;  6253.000
0BF6 0 5 0 0 1 0 0 0 00 0 0EB 0 0 0 0 0 0 0 0  BEB       *GO TO S.P.WRT.RD.STOP;  6254.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          O P R N D . W R T . S T O P
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                  6255.000

                                                                S.P.OPRND.WRT.STOP                                6256.000

08F7 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A        *LINK S.P.FETCH.C;                                     6257.000

08F8 0 0 0 2 0 D 0 0 02 0 040 0 0 0 0 0 0 0 0             NOD=@00000400&T; TEST FOR OPERAND WRT STOP             6258.000

08F9 0 0 0 2 1 E F 0 02 0 040 0 0 0 0 0 0 0 0             T=@00000400; IF NOT OFF TURN ON                        6259.000

08FA C 4 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0  BF0        IF NALUZ *GO TO TURN.OFF.STOP;                         6260.000

08FB 0 5 7 0 1 0 0 0 00 0 0D8 0 0 0 0 0 0 0 0  BD8        *LINK S.P.SET.C.FLAG;                                  6261.000

08FC 0 0 0 0 0 A F 0 02 0 020 0 0 0 0 0 0 0 0             T=@02000000:T; SET OP.WRT.STOP                         6262.000

08FD 0 5 0 0 1 0 0 0 00 0 0EB 0 0 0 0 0 0 0 0  BEB        *GO TO S.P.WRT.RD.STOP;                                6263.000
```

```
                                                                                                    6264.000
                                                              S.P.EXTFUNC                           6265.000
0BFE 0 0 0 0 6 D F 0 02 0 0F0 0 0 0 0 0 0 0 0                      T=@0F000000&PNLDATA;            6266.000
0BFF 0 0 0 0 0 8 0 0 02 0 040 0 0 0 0 0 0 0 0                      NOD=@04000000!T;           EXT FCN 4 ?    6267.000
0C00 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                      *NOP;                           6268.000
0C01 C 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  C0F                 IF NALUZ *GO TO S.P.EXTFUNC.1;  6269.000
0C02 3 5 0 0 1 0 0 0 00 3 03B 0 0 0 0 0 0 0 0  C3B                 IF FFRUN *GO TO S.P.FAULT.2;    6270.000
0C03 0 0 0 7 1 E F 0 00 0 FF0 0 0 0 0 0 0 0 0                      T=@FFFFFFFF;                    6271.000
0C04 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0                      SCRATCH(@89)=T; STORE A         6272.000
0C05 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                      SCRATCH(@8B)=T; STORE B         6273.000
0C06 0 0 0 2 1 9 F 0 02 0 100 0 0 0 0 0 0 0 0                      T=%@00001000;   (T=FFFFFFFF)    6274.000
0C07 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0                      SCRATCH(@88)=T; STORE C         6275.000
0C08 0 4 0 1 0 0 4 0 02 A 8AC 0 0 0 0 0 0 0 0  C0C                 SCRATCH(@8A)=T, *HOP $+4; STORE D  6276.000
                                                              S.P.SET.B                             6277.000
0C09 0 4 0 1 1 E F 0 02 0 40D 0 0 0 0 0 0 0 0  C0D                 T=@00400000, *HOP $+4;          6278.000
                                                              S.P.SET.A.C.D                         6279.000
0C0A 0 4 0 1 1 E F 0 02 0 80D 0 0 0 0 0 0 0 0  C0D                 T=@00800000,*HOP $+3;           6280.000
                                                              S.P.SET.B.C.D                         6281.000
0C0B 0 4 0 1 1 E F 0 02 0 70D 0 0 0 0 0 0 0 0  C0D                 T=@00700000, *HOP $+2;          6282.000
                                                              S.P.SET.ALL.FLAGS                     6283.000
0C0C 0 0 0 1 1 E F 0 02 0 F00 0 0 0 0 0 0 0 0                      T=@00F00000;                    6284.000
                                                              S.P.SAVE.FLAG                         6285.000
0C0D 0 0 0 1 0 0 4 0 02 A 860 0 0 0 0 0 0 0 0                      SCRATCH(@86)=T; STORE TC.FLG.ID 6286.000
0C0E 0 6 0 0 1 0 0 0 00 0 A8D 0 0 0 0 0 0 0 0  A8D                 *GO TO S.P.UART.TBMT;           6287.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU EXTFUNC.1

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6288.000

                                                    S.P.FXTFUNC.1                                                   6289.000

0C0F 0 0 0 0 0 8 0 0 02 0 010 0 0 0 0 0 0 0 0                NOD=@01000000!T; TEST FOR EXTENDED FUNC 1              6290.000

0C10 0 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *NOP;                                                  6291.000

0C11 C 5 0 0 1 0 0 0 00 0 03E 0 0 0 0 0 0 0 0  C3E           IF NALUZ *GO TO S.P.FAULT.5;                           6292.000

0C12 0 0 0 2 1 E F 0 02 0 100 0 0 0 0 0 0 0 0                T=@00001000; SET BIT 19 OF THE 'D' DISPLAY             6293.000

0C13 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0                SCRATCH(@8A)=T; STORE 'D' DISPLAY DATA                 6294.000

0C14 0 6 7 0 1 0 0 0 00 0 A78 0 0 0 0 0 0 0 0  A78           *LINK S.P.FETCH.A;                                     6295.000

0C15 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0                SCRATCH(@89)=T; STORE 'T' IN 'A' DISPLAY LOCATION      6296.000

0C16 0 0 0 3 1 E F 0 02 0 210 0 0 0 0 0 0 0 0                T=@00000021;                SET MA & MD                6297.000

0C17 0 6 7 0 1 0 0 0 00 0 AEE 0 0 0 0 0 0 0 0  AEE           *LINK S.P.C.BYTE3;                                     6298.000

0C18 0 5 0 0 1 0 0 0 00 0 00A 0 0 0 0 0 0 0 0  C0A           *GO TO S.P.SET.A.C.D;                                  6299.000

                                                    EXIT                                                            6300.000

0C19 0 0 0 3 1 E E 0 02 0 004 0 0 0 0 0 0 0 0                FULLMAR=0, RESET(HIREG);                               6301.000

0C1A 0 1 0 0 1 0 0 0 00 0 205 0 0 0 0 0 0 0 0                RESET(UARTDAV) *JUMPJ;                                 6302.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    PSW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6303.000
                                                                 S.P.PSW                                            6304.000
0C1B 0 6 7 0 1 0 0 0 00 0 A7C 0 0 0 0 0 0 0 0  A7C                  *LINK S.P.FETCH.FUNCTION;                          6305.000
0C1C 0 0 0 0 0 3 F 0 02 0 060 0 0 0 0 0 0 0 0                       T=@06000000+T;   IS IT A READ?                     6306.000
0C1D 3 5 0 0 1 0 0 0 00 3 038 0 0 0 0 0 0 0 0  C38                  IF FFRUN *GO TO S.P.FAULT.2;                       6307.000
0C1E 2 5 0 0 1 0 0 0 00 0 723 0 0 0 0 0 0 0 0  C23                  IF ALU4-7Z *GO TO S.P.READ.PSW;                    6308.000
                                                                 S.P.WRITE.PSW                                      6309.000
0C1F 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79                  *LINK S.P.FETCH.B;                                 6310.000
0C20 0 6 0 0 1 E 6 0 00 0 258 0 0 0 0 0 0 0 0  258                  DI=S, *GO TO PPATTN.WT.PSW; GO TO PARALLEL PANEL WRITE PSW  6311.000
                                                                 S.P.PSW.RD                                         6312.000
0C21 0 0 0 1 0 0 4 0 02 A 8D0 0 0 0 0 0 0 0 0                       SCRATCH(@8D)=T; SAVE ADDRESS STOP BITS (T=00000011)  6313.000
0C22 0 0 0 0 1 E D 0 02 8 800 0 0 0 0 0 0 0 0                       NL=@80000000, PCTOMAR;                             6314.000
                                                                 S.P.READ.PSW                                       6315.000
0C23 0 6 7 0 1 0 0 0 00 8 ACE 0 0 0 0 0 0 0 0  ACE                  PCTOMAR, *LINK S.P.MEMORY.READ;                    6316.000
0C24 5 5 0 0 1 0 0 0 00 4 04D 0 0 0 0 0 0 0 0  C4D                  IF %BMUX00 *GO TO S.P.ERROR.3.4.5.7;READ ERROR, EVACUATE.  6317.000
0C25 0 0 0 1 3 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0                       SCRATCH(@8B)=DI;  STORE 'B'                        6318.000
0C26 0 6 7 0 1 0 0 0 00 8 235 0 0 0 0 0 0 0 0  235                  PCTOMAR, *LINK PPATTN.RD.PSW; GO TO PARALLEL PANEL READ PSW  6319.000
                                                                 S.P.SET.PSW                                        6320.000
0C27 0 0 0 1 0 0 4 0 02 A 890 0 0 0 0 0 0 0 0                       SCRATCH(@89)=T;  COMBINED DATA TO 'A' DISPLAY       6321.000
0C28 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0                       T=@00000000;                                       6322.000
0C29 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0                       SCRATCH(@8A)=T; STORE 0                            6323.000
0C2A 0 0 0 1 1 0 1 0 02 A 8D0 0 0 0 0 0 0 0 0                       S=SCRATCH(@8D);  FETCH STOP CODE                   6324.000
0C2B 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0                       T=S;                                               6325.000
0C2C 0 0 0 7 0 3 0 0 00 0 EF0 0 0 0 0 0 0 0 0                       NOD=@FFFFFFEF+T;                                   6326.000
0C2D 0 0 0 3 1 E F 0 02 0 420 0 0 0 0 0 0 0 0                       T=@00000042;                                       6327.000
0C2E B 5 0 0 1 0 0 0 00 0 030 0 0 0 0 0 0 0 0  C30                  IF ALUZ *GO TO $+2;                                6328.000
0C2F 0 6 7 0 1 0 0 0 00 0 AEF 0 0 0 0 0 0 0 0  AEE                  *LINK S.P.C.BYTE3;                                 6329.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    PSW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0C30 0 5 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  C0C       *GO TO S.P.SET.ALL.FLAGS;                                    6330.000

                                                S.P.ADDRESS.STOP                                                     6331.000

0C31 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A       *LINK S.P.FETCH.C;                                           6332.000

0C32 0 0 0 7 0 D F 0 00 0 800 0 0 0 0 0 0 0 0             T=@FFFFFF80&T;                                               6333.000

0C33 0 0 0 3 0 A F 0 02 0 C20 0 0 0 0 0 0 0 0             T=@000000C2:T; SET 'STOP' AND 'PSW' INDICATOR LIGHTS         6334.000

0C34 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0             SCRATCH(@88)=T; STORE IN 'C'                                 6335.000

0C35 0 0 0 1 1 E F 0 02 0 200 0 0 0 0 0 0 0 0             T=@00200000; SET FLAG FOR 'C' DISPLAY                        6336.000

0C36 0 5 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  C0D       *GO TO S.P.SAVE.FLAG;                                        6337.000

                                                STEP.HALT                                                            6338.000

0C37 5 6 0 0 1 0 0 0 00 0 0CE 0 0 0 0 0 0 0 0   CE       IF %EXTLW *GO TO PANEL.HALT;                                 6339.000

0C38 1 4 0 0 1 0 0 0 00 0 7DA 0 0 0 0 0 0 0 0  C3A       IF OPNORESP *HOP $+2;                                        6340.000

0C39 0 6 0 0 1 0 0 0 00 0 0CE 0 0 0 0 0 0 0 0   CE       *GO TO PANEL.HALT;                                           6341.000

0C3A 6 4 0 0 4 A B 7 02 0 019 0 0 0 0 0 0 0 0  C39       FR(TRACE)=@01000000:FR(TRACE), *HOP $-1;                     6342.000

                                                S.P.FAULT.2                                                          6343.000

0C3B 0 4 0 3 1 E F 0 02 0 02F 0 0 0 0 0 0 0 0  C3F       T=@00000002, *HOP $+4;    OPERATION NOT PERMITTED IN RUN MODE 6344.000

                                                S.P.FAULT.3                                                          6345.000

0C3C 0 4 0 3 1 E F 0 02 0 03F 0 0 0 0 0 0 0 0  C3F       T=@00000003, *HOP $+3;    INVALID OPERAND SOURCE OR DESTINATION 6346.000

                                                S.P.FAULT.4                                                          6347.000

0C3D 0 4 0 3 1 E F 0 02 0 04F 0 0 0 0 0 0 0 0  C3F       T=@00000004, *HOP $+2;    INVALID SEQUENCE                   6348.000

                                                S.P.FAULT.5                                                          6349.000

0C3E 0 0 0 3 1 E F 0 02 0 050 0 0 0 0 0 0 0 0             T=@00000005;              INVALID EXTENDED FUNCTION          6350.000

0C3F 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0             SCRATCH(@88)=T;           SAVE FOR DISPLAY B                 6351.000

0C40 0 6 7 0 1 0 0 0 00 0 A7A 0 0 0 0 0 0 0 0  A7A       *LINK S.P.FETCH.C;                                           6352.000

0C41 0 0 0 3 0 A F 0 02 0 080 0 0 0 0 0 0 0 0             T=@00000008:T;            SET OPERATOR FAULT INDICATOR       6353.000

0C42 0 0 0 7 0 D F 0 00 0 0F0 0 0 0 0 0 0 0 0             T=@FFFFFF0F&T;            MASK OUT MA, EA, INST, STOP        6354.000

0C43 0 0 0 1 0 0 4 0 02 A 880 0 0 0 0 0 0 0 0             SCRATCH(@88)=T;           SAVE FOR DISPLAY C                 6355.000

0C44 0 6 7 0 1 0 0 0 00 0 A7B 0 0 0 0 0 0 0 0  A7B       *LINK S.P.FETCH.D;                                           6356.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    PSW
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                  **       ENTER HERE WITH DISPLAY D IN T REG                          6357.000
                                                  S.P.ERROR.FUNNEL                                                    6358.000
0C45 0 0 0 6 0 D F 0 02 0 EF0 0 0 0 0 0 0 0 0             T=@FFFFEFFF&T;          MASK OUT EXTENDED FUNCTION 1        6359.000
0C46 0 0 0 1 0 0 4 0 02 A 8A0 0 0 0 0 0 0 0 0             SCRATCH(@8A)=T;         SAVE FOR DISPLAY D                  6360.000
0C47 0 5 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  C0C        *GO TO S.P.SET.ALL.FLAGS;                                   6361.000
                                                  S.P.ERROR.1                                                         6362.000
0C48 0 0 0 0 1 0 0 0 00 0 A05 0 0 0 0 0 0 0 0             SET(UARTDAV);           RESET ERROR                         6363.000
0C49 0 4 0 3 1 E F 0 02 0 01C 0 0 0 0 0 0 0 0  C4C        T=@00000001, *HOP $+3;    UART ERROR                        6364.000
                                                  S.P.ERROR.2                                                         6365.000
0C4A 0 4 0 3 1 E F 0 02 0 02C 0 0 0 0 0 0 0 0  C4C        T=@00000002, *HOP $+2;    BAD DATA BUT NO UART ERROR REPORTED 6366.000
                                                  S.P.ERROR.6                                                         6367.000
0C4B 0 0 0 3 1 E F 0 02 0 060 0 0 0 0 0 0 0 0             T=@00000006;            WRITE/READ COMPARE ERROR            6368.000
0C4C 0 0 0 1 0 0 4 0 02 A 8B0 0 0 0 0 0 0 0 0             SCRATCH(@8B)=T;         SAVE FOR DISPLAY B                  6369.000
                                                  S.P.ERROR.3.4.5.7                                                   6370.000
0C4D 0 6 7 0 1 0 0 0 00 0 A7B 0 0 0 0 0 0 0 0  A7B        *LINK S.P.FETCH.D;                                          6371.000
0C4E 0 4 0 3 0 A F 0 02 0 015 0 0 0 0 0 0 0 0  C45        T=@00000001:T, *HOP S.P.ERROR.FUNNEL;   SET ERROR INDICATOR 6372.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 75 MODE INTERRUPT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6373.000
                                                    ***                                                             6375.000
                                                    ***     ENTRY PARAMETER :                                       6376.000
                                                    ***       T = R(N.PSW2)                                         6377.000
                                                    ***                                                             6378.000
                                                                                                                    6379.000
                                                    (@C4F)                                                          6380.000
                                                    HANDLE.75.INT                                                   6381.000
0C4F 0 0 0 0 4 0 F 0 00 0 000 0 0 0 0 0 0 0 0               T=R(N.PSW2);   FETCH NEW PSW                            6382.000
0C50 5 5 0 0 0 0 0 0 00 8 068 0 0 0 0 0 0 0 0  C68         NOD=T, IF %BMUX16 *GO TO USE.CURRENT.PSW; TEST BIT 16 OF PSW 2  6383.000
0C51 0 0 0 0 0 0 0 0 00 0 906 0 0 0 0 0 0 0 0               NOD=T, SET(UNBLOCK);                                    6384.000
0C52 5 0 0 0 0 0 0 0 00 9 106 0 0 0 0 0 0 0 0               NOD=T, IF %BMUX17 RESET(UNBLOCK);                       6385.000
0C53 3 1 0 0 0 0 0 0 00 9 000 0 0 0 0 0 0 0 0               NOD=T, IF BMUX17 *JUMPJ;    EXIT IF UNBLOCK             6386.000
                                                    *                 IS SELECTED BY N.PSW2                         6387.000
                                                    FAST.DEACTIVATE                                                 6388.000
0C54 6 0 0 1 4 D 0 3 02 0 800 0 0 0 0 0 0 0 0               NOD=@00800000&FR(INTRTAB); TEST FOR RTOM INT ENTRY      6389.000
0C55 0 0 0 3 3 E F 0 02 0 080 0 0 0 0 0 0 0 0               T=@00000008, BMUX=DI; SET DFACTIVATE BUS CODE AND WAIT FOR  6390.000
                                                    *             ACKNOWLEDGE DRT                                   6392.000
0C56 C 5 0 0 0 1 C 0 00 F 062 0 0 0 0 0 0 0 0  C62         NU=%T(ZE), IF NALUZ *GO TO ICT.CHECK;SKIP AICT IF RTOM INTERRUPT  6394.000
                                                    AICT.DEACT                                                      6395.000
0C57 9 4 0 0 1 0 0 0 00 0 DB9 0 0 0 0 0 0 0 0  C59         IF %IOTIMEOUT *HOP $+2;                                 6396.000
0C58 0 6 0 0 1 0 0 0 00 0 52D 0 0 0 0 0 0 0 0  52D         *GO TO INT.DRT.TIMEOUT;  EXIT TO INTERRUPT I/O ERROR    6397.000
0C59 0 6 7 0 1 0 0 0 00 0 307 0 0 0 0 0 0 0 0  307         NOD=MAR, *LINK CMD.AICT;  GO ISSUE AICT                 6398.000
0C5A 9 4 3 0 1 0 0 0 00 0 ABE 0 0 0 0 0 0 0 0  C5E         DECRN, IF %IORETRY:IOCHBUSY *HOP DEACT.WAIT.READY;      6399.000
0C5B 0 6 7 0 1 0 0 0 00 0 30C 0 0 0 0 0 0 0 0  30C         *LINK TEST.RETRY;                                       6400.000
0C5C A 4 0 0 1 0 0 0 00 0 907 0 0 0 0 0 0 0 0  C57         IF %NCTRZ *GO TO AICT.DEACT; RETRY 256 TIMES            6401.000
                                                    INTR.ERR6                                                       6402.000
0C5D 0 6 0 0 1 0 0 0 00 0 520 0 0 0 0 0 0 0 0  520         *GO TO INT.IO.ERR1;                                     6403.000
                                                    DEACT.WAIT.READY                                                6404.000
0C5E 1 4 0 0 0 1 C 0 00 F 58D 0 0 0 0 0 0 0 0  C5D         NU=%T(ZE), IF IONORESP:IOTIMEOUT *GO TO INTR.ERR6;      6405.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 75 MODE INTERRUPT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                *              ; SET READY TIMEOUT FOR 512 CLOCKS (76.8 US)                6406.000
0C5F 1 4 0 0 1 0 0 0 00 0 7A4 0 0 0 0 0 0 0 0  C64            IF IORESPRDY *HOP ICT.DFACT; EXIT IF READY RECEIVED         6407.000
0C60 A 5 3 0 1 0 0 0 00 0 95F 0 0 0 0 0 0 0 0  C5F            DECRN, IF %NCTR7 *GO TO $-1;  LOOP FOR READY AND DECREMENT COUNT 6408.000
                                                INTR.ERR7                                                                   6409.000
0C61 0 6 0 0 1 0 0 0 00 0 52C 0 0 0 0 0 0 0 0  52C            *GO TO INT.RDY.TIMEOUT;                                      6410.000
                                                ICT.CHECK                                                                   6411.000
0C62 9 4 0 0 1 0 0 0 00 0 DB4 0 0 0 0 0 0 0 0  C64            IF %IOTIMEOUT *HOP ICT.DEACT;                                6412.000
0C63 0 6 0 0 1 0 0 0 00 0 52D 0 0 0 0 0 0 0 0  520            *GO TO INT.DRT.TIMEOUT; EXIT TO INTERRUPT I/O ERROR          6413.000
                                                ICT.DEACT                                                                   6414.000
0C64 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0                 NU=@0F000000;  SET RETRY COUNT TO 16                         6415.000
0C65 0 6 7 0 1 0 1 0 00 0 308 0 0 0 0 0 0 0 0  308            S=N, *LINK CMD.ICT; SAVE RETRY COUNT & GO ISSUE ICT          6416.000
0C66 9 4 0 0 0 0 C 0 00 F 28F 0 0 0 0 0 0 0 0  C6F            NU=T(ZE), IF %IONORESP:IOCHBUSY:IORETRY *HOP EXIT.USE.CURRENT; 6417.000
0C67 9 4 0 0 1 E C 0 00 0 EBA 0 0 0 0 0 0 0 0  C6A            NU=S, IF %IORETRY *HOP INTR.ERR8;                            6418.000
0C68 0 6 7 0 1 0 0 0 00 0 30C 0 0 0 0 0 0 0 0  30C            *LINK TEST.RETRY; GO TEST RETRY;                             6419.000
0C69 A 4 3 0 1 0 0 0 00 0 905 0 0 0 0 0 0 0 0  C65            DECRN, IF %NCTR7 *GO TO ICT.DFACT+1; RETRY 16 TIMES          6420.000
                                                INTR.ERR8                                                                   6421.000
0C6A 0 6 0 0 1 0 0 0 00 0 529 0 0 0 0 0 0 0 0  529            *GO TO INT.IO.ERR2;                                          6422.000
                                                USE.CURRENT.PSW                                                             6423.000
0C6B 0 0 0 0 4 0 0 0 00 D F00 0 0 0 0 0 0 0 0                 NOU=R(TRACE), OTHERBANK;                                     6424.000
0C6C 5 1 0 0 1 0 0 0 00 9 000 0 0 0 0 0 0 0 0                 IF %BMUX17 *JUMPJ;                                           6425.000
0C6D 0 0 0 0 1 0 0 0 00 0 106 0 0 0 0 0 0 0 0                 RESET(UNBLOCK);                                              6426.000
0C6E 0 5 0 0 1 0 0 0 00 0 054 0 0 0 0 0 0 0 0  C54            *GO TO FAST.DEACTIVATE;                                      6427.000
                                                EXIT.USE.CURRENT                                                            6428.000
0C6F 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                 *JUMPJ;                                                      6429.000
                                                S.P.IPL                                                                     6430.000
0C70 0 6 7 0 1 0 0 0 00 0 A79 0 0 0 0 0 0 0 0  A79            *LINK S.P.FETCH.B;                                           6431.000
0C71 0 0 0 0 0 0 0 0 00 0 005 0 0 0 0 0 0 0 0                 NOU=T, RESET(ENAUORD);                                       6432.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 75 MODE INTERRUPT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0C72 0 6 0 0 1 0 0 0 00 0 168 0 0 0 0 0 0 0 0  168        *GO TO SAVE.IPL.ADDR;  RETURN TO MAIN IPL FLOW                  6433.000
                                                *                                                                      6434.000
                                                (@C73)                                                                 6435.000
                                                *                                                                      6436.000
                                                *        CHECK FOR IPU OPERATION--MACHINE TRAP HALT NOT VALID FOR IPU   6437.000
                                                *                                                                      6438.000
                                                IPU.TRAP.CHECK                                                         6439.000
                                                *                                                                      6440.000
0C73 4 6 0 0 1 0 0 0 00 2 F8A 0 0 0 0 0 0 0 0  F8A        IF IPU *GOTO IPU.TRAP.FLAG.TEST;                               6441.000
0C74 0 0 0 5 0 D 8 0 00 0 200 0 0 0 0 0 0 0 0              R(CPSTS)=STATUSRT;        SAVE PRIV/EXT INDEX BITS FOR PSW     6442.000
0C75 0 6 0 0 1 0 0 0 00 0 5A4 0 0 0 0 0 0 0 0  5A4        *GOTO M.75.TRAP.RTN;                                           6443.000
                                                                                                                       6444.000
                                                *                                                                      6445.000
                                                *        THE BEI INSTRUCTION IS NOT VALID IN THE IPU MODE.              6446.000
                                                *                                                                      6447.000
                                                BEI.TEST                                                               6448.000
                                                *                                                                      6449.000
0C76 4 6 0 0 1 0 0 0 00 2 7F5 0 0 0 0 0 0 0 0  7F5        IF IPU *GOTO UNDEF.75;    INSTRUCTION UNDEFINED                6450.000
0C77 0 6 7 0 1 0 0 0 00 0 7F0 0 0 0 0 0 0 0 0  7F0        *LINK CHECK.CPU.MODE;                                          6451.000
0C78 0 6 0 0 1 0 0 0 00 0 7EA 0 0 0 0 0 0 0 0  7EA        *GOTO BEI.1;              CONTINUE EO PROCESSING               6452.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                    M A P   C O N T E X T   S W I T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                6453.000

                                                                                                                6454.000
                                                    ***                                                         6455.000
                                                    ***     INITIALIZE LOAD MAP                                 6456.000
                                                    ***                                                         6457.000
                                                    ***     ENTRY PARAMETER :                                   6458.000
                                                    ***       R(N.PSW2) = NEW PROGRAM STATUS WORD 2             6459.000
                                                    ***                                                         6460.000

                                                                                                                6461.000
                                                    (@C79)                                                      6462.000
                                                    INITIALIZE.LOAD.MAP                                         6463.000
0C79 0 0 0 6 1 6 1 0 02 0 BF0 0 0 0 0 0 0 0 0                S=@FFFFBFFF+1; SET S = @FFFFC000 (CPIX MASK)        6464.000
0C7A 0 5 7 0 4 B F 0 00 D 09F 0 0 0 0 0 0 0 0  C9F           T=%S&R(N.PSW2), OTHERBANK, *LINK FETCH.MSCD; LOAD CPIX IN T  6465.000
0C7B 6 0 0 2 1 E B 6 02 0 800 0 0 0 0 0 0 0 0                FR(BUMP)=@00008000; INITIALIZE MAP REG. ADDR INCREMENT VALUE  6466.000
0C7C 6 0 0 6 1 6 B 3 02 0 7F0 0 0 0 0 0 0 0 0                FR(MAP.ADDR)=@FFFF7FFF+1; SET INITIAL MAP REGISTER ADDR =-1  6467.000
0C7D 6 0 0 4 3 D B 2 02 0 6F0 0 0 0 0 0 0 0 0                FR(MSCD)=@6FFFFFFF&DI; SAVE MAP SEGMENT CONTROL    6468.000
                                                    *                       DESCRIPTOR (MSCD).                  6469.000
                                                    FETCH.MSD1                                                  6470.000
0C7E 5 5 0 0 3 0 E 0 00 4 081 0 0 0 0 0 0 0 0  C81           FULLMAR=DI, IF %BMUX00 *GO TO BORROW.SEGMENTS; LOAD MAP SEGMENT  6471.000
                                                    *                       DESCRIPTOR (MSD) ADDRESS IN MAR.    6472.000
                                                    (@C7F)                                                      6473.000
0C7F 0 4 7 0 0 0 C 0 1C 1 088 0 0 0 0 0 0 0 0  C88           NU=T, READ, FRCWORD, *LINK LOAD.MAP; LOAD SEGMENT DESCRIPTOR  6474.000
                                                    *                COUNT IN N REG AND FETCH MSD.              6475.000
0C80 0 5 0 0 1 E 8 0 00 0 B92 0 0 0 0 0 0 0 0  C92           R(MAP.ADDR)=S, *GO TO CHECK.MAP.CLEAR; SAVE CURRENT MAP REG.  6476.000
                                                    *                       ADDRESS IN T.                       6477.000
                                                    BORROW.SEGMENTS                                             6478.000
0C81 0 0 0 6 1 6 1 0 02 0 BF0 0 0 0 0 0 0 0 0                S=@FFFFBFFF+1; SET S =@FFFFC000 (BPIX MASK)         6479.000
0C82 0 5 7 0 5 B F 0 00 D 09F 0 0 0 0 0 0 0 0  C9F           T=%S&R(N.PSW2,HWS), OTHERBANK, *LINK FETCH.MSCD; LOAD BPIX IN T  6480.000
0C83 0 0 0 4 3 D E 0 02 0 6FD 0 0 0 0 0 0 0 0                FULLMAR=@6FFFFFFF&DI,CLDNU;LOAD MSD ADDR IN MAR &MSD COUNT IN N  6481.000
0C84 0 4 7 0 1 0 0 0 1C 1 088 0 0 0 0 0 0 0 0  C88           READ, FRCWORD, *LINK LOAD.MAP; FETCH BORROWED MSD   6482.000
```

```
                SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                MAP CONTEXT SWITCH
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0C85 0 0 0 0 4 0 6 0 00 0 A00 0 0 0 0 0 0 0 0          DT=R(MSCD); GET INITIAL MAP SEGMENT CONTROL DESCRIPTOR          6483.000

0C86 0 0 0 0 1 E 8 0 00 0 B00 0 0 0 0 0 0 0 0          R(MAP.ADDR)=S; SAVE CURRENT MAP REGISTER ADDRESS                6484.000

0C87 0 5 0 0 3 0 F 0 00 0 07E 0 0 0 0 0 0 0 0  C7E     T=DI, *GO TO FETCH.MSD1;                                        6485.000

                                                                                                                      6486.000
                                                ***                                                                   6487.000
                                                ***    LOAD MAP                                                       6488.000
                                                ***                                                                   6489.000
                                                ***    ENTRY PARAMETERS :                                             6490.000
                                                ***      MAR = MAP SEGMENT DESCRIPTOR ADDRESS                         6491.000
                                                ***      N = MAP SEGMENT CONTROL DESCRIPTOR COUNT                     6492.000
                                                ***      MEMORY READ OF MAP SEGMENT DESCRIPTOR IS IN PROGRESS         6493.000
                                                ***                                                                   6494.000
                                                                                                                      6495.000
                                                LOAD.MAP                                                              6496.000

0C88 9 4 0 0 1 0 0 0 00 0 55A 0 0 0 0 0 0 0 0  C8A     IF %OPNORESP:OPTIMEOUT:OPRNDPE *HOP $+2;                        6497.000

0C89 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530     NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERROR       6498.000

0C8A 0 0 0 0 1 0 8 0 00 0 400 0 0 0 0 0 0 0 0          R(MSDLP)=MAR; SAVE INITIAL SEGMENT DESCRIPTOR COUNT AND POINTER 6499.000

0C8B 0 0 0 0 3 0 0 0 00 D 000 0 0 0 0 0 0 0 0          NOD=DI,OTHERBANK; WAIT FOR DRT                                  6500.000

0C8C 6 1 0 0 4 0 1 3 00 0 900 0 0 0 0 0 0 0 0          S=FR(MAP.ADDR), IF NCTRZ *JUMPJ; LOAD INITIAL MAP REGISTER       6501.000
                                                *      ADDRESS (MINUS 1) AND EXIT IF INITIAL MSCD COUNT =0.            6503.000

0C8D 0 0 0 0 3 0 8 0 00 0 E0D 0 0 0 0 0 0 0 0          R(MSD)=DI, CLDNU; SAVE MAP SEGMENT DESCRIPTOR AND PAGE COUNT    6505.000

0C8E 0 5 7 0 1 0 0 0 00 D 0A4 0 0 0 0 0 0 0 0  CA4     OTHERBANK, *LINK EXECUTE.LOAD.MAP;GO PROCESS THIS MSD           6506.000

                                                                                                                      6507.000
                                                ***                                                                   6508.000
                                                ***    RETURN.PARAMETERS :                                            6509.000
                                                ***      T = MAP SEGMENT DESCRIPTOR LIST POINTER                      6510.000
                                                ***      N = MAP SEGMENT CONTROL DESCRIPTOR COUNT                     6511.000
                                                ***                                                                   6512.000
                                                                                                                      6513.000

0C8F 0 0 0 3 0 3 F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004+T; INCREMENT MSD LIST POINTER                       6514.000

0C90 2 1 0 0 0 0 E 0 00 0 900 0 0 0 0 0 0 0 0          FULLMAR=T, IF NCTRZ *JUMPJ; EXIT IF MSD COUNT =0                6515.000

0C91 0 5 3 0 1 E 8 0 1C 1 B88 0 0 0 0 0 0 0 0  C88     R(MAP.ADDR)=S, DECRN, READ, FRCWORD, *GO TO LOAD.MAP; SAVE      6516.000
                                                *              CURRENT MAP ADDR, FETCH NEXT MSD, AND DECREMENT COUNT.  6518.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MAP CONTEXT SWITCH

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                6520.000
                                                    ***                                                         6522.000
                                                    ***     CHECK CLEAR MAP                                     6523.000
                                                    ***                                                         6524.000
                                                    ***     THIS SUB-ROUTINE PROVIDES A EXIT PATH FOR INITIALIZE LOAD MAP.  6525.000
                                                    ***     THIS SUB-ROUTINE TEST FOR MORE THAN 32 REGISTERS LOADED BY  6526.000
                                                    ***     THE LOAD MAP ROUTINE AND EXITS TO MAP ERROR IS THIS CONDITION 6527.000
                                                    ***     IS DETECTED.  THIS SUB-ROUTINE ALSO CLEARS ANY MAP REGISTERS 6528.000
                                                    ***     NOT LOADED BY THE LOAD MAP ROUTINE.                 6529.000
                                                    ***                                                         6530.000

                                                                                                                6531.000

                                                    CHECK.MAP.CLEAR                                             6532.000

0C92 0 0 0 1 0 D 0 0 02 0 F00 0 0 0 0 0 0 0 0               NOD=@00F00000&T; TEST FOR MAP REGISTER ADDRESS >31.  6533.000

0C93 0 0 0 1 1 E F 0 00 0 000 0 0 0 0 0 0 0 0               T=SLEFT; MOVE MAP REG ADDR TO T BITS 11/15          6534.000

0C94 C 4 0 1 0 C D 0 02 0 1FD 0 0 0 0 0 0 0 0  C9D          NL=@001F0000&%T, IF NALUZ *HOP SET.MAP.N.FOUND; LOAD  6535.000
                                                    *       ONES-COMPLEMENT OF MAP REG ADDRESS IN N REG AND GO TO ERROR 6537.000
                                                    *       IF MAP ADDRESS >31.                                 6538.000

0C95 0 0 0 0 4 0 1 0 00 0 B00 0 0 0 0 0 0 0 0               S=R(MAP.ADDR); GET CURRENT MAP REG ADDR IN S REG     6540.000

0C96 0 0 0 0 0 1 6 0 00 F 805 0 0 0 0 0 0 0 0               DI=%T(ZE), SET(ENAUORD);  SET MAP IMAGE = EFFECTIVE ZEROS  6541.000

                                                    LOOP.CLEAR.MAP                                              6542.000

0C97 E 0 0 0 4 3 1 6 00 0 90E 0 0 0 0 0 0 0 0               S=S+FR(BUMP), IF %NCTRZ CDECRN; INCREMENT MAP ADDRESS AND DECRE- 6543.000

                                                    *                MENT CLEAR COUNT (N-REG)                   6544.000

0C98 2 4 0 0 1 E E 0 00 0 90A 0 0 0 0 0 0 0 0  C9A          FULLMAR=S, IF NCTRZ *GO TO EXIT.CLEAR;               6545.000

0C99 0 4 0 0 3 0 0 0 00 5 007 0 0 0 0 0 0 0 0  C97          NOD=DI, LDMAP, *GO TO LOOP.CLEAR.MAP; CLEAR A MAP REGISTER  6546.000

                                                    EXIT.CLEAR                                                  6547.000

0C9A 0 0 0 0 4 0 F 0 00 D 000 0 0 0 0 0 0 0 0               T=R(N.PSW2), OTHERBANK;  FETCH NEW PSD 2            6548.000

0C9B 0 0 0 3 1 E E 0 02 0 005 0 0 0 0 0 0 0 0               FULLMAR=@00000000, RESET(ENAUORD); CLEAR MAR        6549.000

0C9C 0 1 0 1 0 0 4 0 02 A 900 0 0 0 0 0 0 0 0               SCRATCH(@90)=T, *JUMPJ; SAVE NEW PSD 2 AND EXIT     6550.000

                                                    SET.MAP.N.FOUND                                             6551.000

0C9D 0 0 0 2 1 E 1 0 02 0 800 0 0 0 0 0 0 0 0               S=MAP.N.FOUND.FLG;  SET NOT FOUND OR OVERFLOW ERROR FLAG  6552.000

0C9E 0 6 0 0 1 0 0 0 00 0 57C 0 0 0 0 0 0 0 0  57C          *GO TO SYSTEM.CHECK.TRAP;                           6553.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    M A P  C O N T E X T  S W I T C H
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6554.000

                                                                                                                    6555.000
                                                ***                                                                 6556.000
                                                *** FETCH MAP SEGMENT CONTROL DESCRIPTOR (MSCD)                      6557.000
                                                ***                                                                 6558.000
                                                ***    ENTRY PARAMETER :                                            6559.000
                                                ***      T = MASTER PROCESS LIST INDEX (CPIX OR BPIX)               6560.000
                                                ***                                                                 6561.000

                                                                                                                    6562.000

                                                FETCH.MSCD                                                          6563.000

0C9F 0 0 0 1 1 0 1 0 02 A 830 0 0 0 0 0 0 0 0           S=SCRATCH(@83); GET MASTER PROCESS LIST (MPL) BASE ADDRESS  6564.000

0CA0 0 0 0 0 0 3 E 0 00 0 000 0 0 0 0 0 0 0 0           FULLMAR=S+T; ADD BASE ADDRESS AND INDEX VALUES              6565.000

0CA1 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0           READ, FRCWORD; FETCH MAP SEGMENT CONTROL DESCRIPTOR         6566.000

0CA2 9 1 0 0 1 0 0 0 00 0 550 0 0 0 0 0 0 0 0           IF %OPNORESP:OPTIMEOUT:OPRNDPE *JUMPJ; RETURN IF NO MEM ERRORS  6567.000

0CA3 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530      NU=T(ZE), *GO TO CURRENT.INST.ERROR; EXIT TO MEMORY ERRORS  6568.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  79APR27
S E L  32/75  C P U                                                  M A P   C O N T E X T   S W I T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6569.000
                                                    ***                                                             6571.000
                                                    ***     EXECUTE LOAD MAP                                        6572.000
                                                    ***                                                             6573.000
                                                    ***     ENTRY PARAMETERS :                                      6574.000
                                                    ***       N = MAP SEGMENT PAGE COUNT                            6575.000
                                                    ***       R(MSD) = CURRENT MAP SEGMENT DESCRIPTOR               6576.000
                                                    ***       R(BUMP) = MAP REGISTER ADDRESS BUMP VALUE (@00008000) 6577.000
                                                    ***       S = CURRENT MAP REGISTER ADDRESS MINUS 1, IN BITS 12/16 6578.000
                                                    ***       R(MSDLP) = CURRENT MAP SEGMENT DESCRIPTOR LIST POINTER 6579.000
                                                    ***                                                             6580.000

                                                                                                                    6581.000
                                                    EXECUTE.LOAD.MAP                                                6582.000
0CA4 2 5 0 0 1 0 0 0 00 0 9B0 0 0 0 0 0 0 0 0  CB0          IF NCTRZ *GO TO EXIT.LOAD+1; EXIT IF INITIAL MSD COUNT =0.  6583.000
                                                    LOOP.LOAD.MAP                                                   6584.000
0CA5 0 0 0 0 4 0 E 0 00 D E00 0 0 0 0 0 0 0 0               FULLMAR=R(MSD), OTHERBANK; FETCH CURRENT MSD               6585.000
0CA6 0 0 0 0 4 3 1 0 1C 0 F82 0 0 0 0 0 0 0 0               S=S+R(BUMP), READ,FORCEZ; BUMP MAP REG ADDR & GET MAP IMAGE 6586.000
0CA7 9 4 0 0 1 0 0 0 00 0 559 0 0 0 0 0 0 0 0  CA9          IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2; TEST FOR PREVIOUS ERROR 6587.000
0CA8 0 6 0 0 0 0 C 0 00 F 530 0 0 0 0 0 0 0 0  530          NU=T(ZE), *GO TO CURRENT.INST.ERROR;                       6588.000
0CA9 6 0 0 3 4 3 B 6 02 D 040 0 0 0 0 0 0 0 0               FR(MSD)=4+FR(MSD), OTHERBANK; BUMP CURRENT MIDLP           6589.000
0CAA 0 0 3 0 1 E E 0 00 D 805 0 0 0 0 0 0 0 0               FULLMAR=S, DECRM, OTHERBANK, SET(ENAUORD); SET MAP REG. ADDRESS 6590.000
                                                    *       MAR 4-8 AND DECREMENT MAP SEGMENT PAGE COUNT.           6592.000
0CAB 0 0 0 0 4 3 1 0 00 0 E00 0 0 0 0 0 0 0 0               S=S+R(BUMP); INCREMENT TO NEXT MAP REG. ADDRESS            6594.000
0CAC 6 4 0 0 3 1 B 5 00 5 90F 0 0 0 0 0 0 0 0  CAF          FR(MID)=%DI,LDMAP,IF NCTRZ *GOTO EXIT.LOAD; LOAD MAP FROM BITS 6595.000
                                                    *               01/15 AND EXIT IF SEGMENT PAGE COUNT COUNT =0.  6597.000
0CAD 0 0 3 0 1 E E 0 00 0 000 0 0 0 0 0 0 0 0               FULLMAR=S, DECRM; SET NEXT MAP REG. ADDRESS IN MAR BITS    6599.000
                                                    *               4-8 AND DECREMENT SEGMENT PAGE COUNT COUNT.     6601.000
0CAE E 4 0 0 5 0 0 5 00 5 905 0 0 0 0 0 0 0 0  CA5          NOD=FR(MID,HWS),LDMAP,IF %NCTRZ *GOTO LOOP.LOAD.MAP; LOAD THE MAP 6603.000
                                                    *       FROM BITS 01/15 AND LOOP IF SEGMENT PAGE COUNT COUNT NOT =0. 6605.000
                                                    EXIT.LOAD                                                       6607.000
0CAF 0 0 0 0 1 0 1 0 00 0 005 0 0 0 0 0 0 0 0               S=MAR, RESET(ENAUORD); RESTORE CURRENT MAP ADDR IN S       6608.000
0CB0 0 1 0 0 4 0 F 0 00 D 40D 0 0 0 0 0 0 0 0               T=R(MSDLP), OTHERBANK, CLDNU, *JUMPJ; LOAD CURRENT MAP     6609.000
                                                    *       SEGMENT LIST POINTER IN T AND SEGMENT DESCRIPTOR COUNT IN 6611.000
                                                    *       N COUNTER, THEN EXIT EXECUTE LOAD.MAP.                  6612.000
                                                                                                                    6613.000
                                                    ***                                                             6614.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    M A P   C O N T E X T   S W I T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    ***    LOAD PSEUDO MAP                                                    6615.000
                                                    ***                                                                       6616.000
                                                    ***    THIS SUB-ROUTINE LOADS A ONE TO ONE MAP IMAGE OF THE FIRST         6617.000
                                                    ***    128K WORDS OF MEMORY IN MAP REGSITERS 00-0F.  A VALID BIT IS       6618.000
                                                    ***    PROVIDED FOR EACH MAP REGISTER ENTRY.  MAP REGISTERS 10-1F ARE     6619.000
                                                    ***    LOAD WITH A ONE TO ONE IMAGE OF THE FIRST 128K WORDS OF MEMORY,    6620.000
                                                    ***    HOWEVER, NO VALID BIT IS SET.                                      6621.000
                                                    ***                                                                       6622.000

                                                                                                                              6623.000

                                                    LOAD.PSEUDO.MAP                                                           6624.000

0CB1 6 0 0 2 1 E B 6 02 0 800 0 0 0 0 0 0 0 0                       FR(BUMP)=@00008000; SET MAP REG ADDR BUMP VALUE                6625.000

0CB2 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0                       NU=@0F000000;                                                 6626.000

0CB3 0 0 0 6 1 6 E 0 02 0 7F0 0 0 0 0 0 0 0 0                       FULLMAR=@FFFF7FFF+1; SET MAP REG ADDR 0 (-1)                  6627.000

0CB4 0 0 0 5 1 6 1 0 02 0 FE0 0 0 0 0 0 0 0 0                       S=@FFFEFFFF+1; SET MAP IMAGE BUMP VALUE (2-COMP)               6628.000

0CB5 0 4 7 0 1 E F 0 02 0 309 0 0 0 0 0 0 0 0  CB9                  T=@30000000, *LINK LD.PSEUDO.MAP; SET INITIAL MAP IMAGE       6629.000
                                                    *                       ( HVALID =1, HWRPM =0 )                           6631.000

0CB6 0 0 0 0 1 E C 0 02 0 0F0 0 0 0 0 0 0 0 0                       NU=@0F000000; SET EXT MAP COUNT                               6633.000

0CB7 0 4 7 0 1 E F 0 02 0 709 0 0 0 0 0 0 0 0  CB9                  T=@70000000, *LINK LD.PSEUDO.MAP; SET INITIAL EXT MAP IMAGE   6634.000
                                                    *                       ( HVALID =0, HWRPM =0 )                           6636.000

0CB8 0 1 0 0 1 0 0 0 00 0 007 0 0 0 0 0 0 0 0                       *JUMPJ, CLRS;                                                 6638.000

                                                    LD.PSEUDO.MAP                                                             6639.000

0CB9 0 0 0 0 1 0 0 0 00 0 805 0 0 0 0 0 0 0 0                       SET(ENAUORD);                                                 6640.000

                                                    LOOP.PSEUDO                                                               6641.000

0CBA 0 0 0 4 1 3 E 0 00 0 F00 0 0 0 0 0 0 0 0                       FULLMAR=R(BUMP)+MAR; LOAD MAP REG ADDR IN MAR 4-8             6642.000

0CBB A 4 3 0 0 3 F 0 00 5 90A 0 0 0 0 0 0 0 0  CBA                  T=S+T, DECRN, LDMAP, IF %NCTRZ *GO TO LOOP.PSEUDO;             6643.000

                                                    EXIT.LD.PSEUDO                                                            6644.000

0CBC 0 1 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0                       RESET(ENAUORD), *JUMPJ;                                       6645.000

                                                                                                                              6646.000
                                                    ***                                                                       6647.000
                                                    ***  TRANSFER MAP TO REGISTER -TMAPR-, SECONDARY DECODE (Q=0B)             6648.000
                                                    ***  OP CODE = 2C                                                         6649.000
                                                    ***                                                                       6650.000

                                                                                                                              6651.000

                                                    TMAPR                                                                     6652.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU MAP CONTEXT SWITCH

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0CBD 0 6 7 0 1 0 0 0 00 0 7F0 0 0 0 0 0 0 0 0  7F0        *LINK CHECK.CPU.MODE; GO VERIFY 75 MODE                          6653.000
0CBE 0 0 0 1 5 A 1 5 02 0 01D 0 0 0 0 0 0 0 0              S=@00010000:R(S,HWS), CLDNU;GET MAP ADDR IN 11-15 AND SET LSB    6654.000
                                                *                  OF THE MAP REGISTER ADDRESS (BIT 16)                  6655.000
0CBF 0 0 0 0 1 0 0 0 01 0 080 0 0 0 0 0 0 0 0              SHIFTS(SRL); SHIFT MAP ADDRESS TO BITS 12-16                    6656.000
                                                MAPADD                                                                  6657.000
0CC0 9 4 0 0 1 E E 0 00 0 F72 0 0 0 0 0 0 0 0  CC2        FULLMAR=S, IF %MAPMODE *HOP $+2;  TMAPR IS ILLEGAL IN MAP MODE  6658.000
0CC1 0 6 0 0 1 0 0 0 00 0 F33 0 0 0 0 0 0 0 0  F33        *GO TO MAP.MODE.INVALID.ERR+1; SET ERR+1 FLG & SYSTEM CHECK TRAP 6659.000
0CC2 0 0 0 5 1 E 8 0 00 0 B05 0 0 0 0 0 0 0 0              R(TEMP1)=STATUS, SET(MAPMODE); SAVE CURRENT CC'S AND EXT BIT     6660.000
0CC3 0 0 0 0 1 E 1 0 02 0 040 0 0 0 0 0 0 0 0              S=@04000000;  SET EXT BIT IN S                                  6661.000
0CC4 0 0 0 0 1 0 0 0 00 0 50C 0 0 0 0 0 0 0 0              SETXCC(S); ENTER EXTENDED INDEXING                              6662.000
0CC5 0 0 0 2 1 E F 0 02 0 805 0 0 0 0 0 0 0 0              T=@00008000, SET(ENAUORD);                                      6663.000
0CC6 1 4 0 0 1 0 3 0 00 0 8D6 0 0 0 0 0 0 0 0  CC6        MAR=MAR, IF BIBUSY *GO TO $ ; MAP THE ADDR IN MAR               6664.000
0CC7 0 0 0 1 0 A 1 0 02 0 FF0 0 0 0 0 0 0 0 0              S=@00FF0000:T; SET MAP CONTENTS MASK TO @00FF8000               6665.000
0CC8 0 0 0 0 1 D 9 0 00 6 000 0 0 0 0 0 0 0 0              R(R)=S&MAR,RDMAP; READ ODD MAP ENTRY INTO BITS 08-16             6666.000
0CC9 0 0 0 1 1 E 6 0 02 0 600 0 0 0 0 0 0 0 0              DI=@00600000; SET VALID & WRITE PROTECT BIT MASK                 6667.000
0CCA 0 0 0 5 3 D F 0 00 0 000 0 0 0 0 0 0 0 0              T=STATUS&DI; GET ODD REGISTER VALID & WRITE PROTECT BITS         6668.000
0CCB 0 0 0 1 1 D 3 0 02 0 0F0 0 0 0 0 0 0 0 0              MAR=@000F0000&MAR; CLEAR MAP ADDR LSB AND REMAP                 6669.000
0CCC 6 0 0 1 0 8 B 5 02 0 400 0 0 0 0 0 0 0 0              ER(TEMP3)=@00400000!T; INVERT THE LVALID BIT                     6670.000
0CCD 0 0 0 0 1 D 1 0 00 6 000 0 0 0 0 0 0 0 0              S=S&MAR,RDMAP; READ EVEN MAP ENTRY INTO BITS 08-16               6671.000
0CCE 0 0 0 0 5 A 1 4 00 0 005 0 0 0 0 0 0 0 0              S=S:R(R,HWS), RESET(ENAUORD); COMBINE EVEN & ODD MAP ENTRIES     6672.000
0CCF 0 0 0 0 1 0 0 0 01 0 010 0 0 0 0 0 0 0 0              SHIFTS(SLC); SHIFT EVEN MAP TO BITS 07-15 & ODD MAP TO 23-31     6673.000
0CD0 0 0 0 0 1 E 9 0 00 0 000 0 0 0 0 0 0 0 0              R(R)=S;                                                         6674.000
0CD1 0 0 0 5 3 D F 0 00 0 000 0 0 0 0 0 0 0 0              T=STATUS&DI; FETCH EVEN MAP VALID & WRITE PROTECT BITS IN 09&10  6675.000
0CD2 0 0 0 1 0 8 1 0 02 0 400 0 0 0 0 0 0 0 0              S=@00400000!T;  INVERT THE LVALID BIT                           6676.000
0CD3 0 0 0 0 5 A F 0 00 0 D00 0 0 0 0 0 0 0 0              T=S:R(TEMP3,HWS); COMBINE EVEN MAP VALID & WRITE PROTECT        6677.000
                                                *            BITS (BITS 09/10) WITH ODD MAP VALID & PROTECT BITS           6678.000
                                                *           (BITS 25/26).                                                  6679.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                    M A P   C O N T E X T   S W I T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0CD4 6 0 0 0 4 0 1 3 00 2 305 0 0 0 0 0 0 0 0          S=FR(TEMP1), TNIBL, RESET(MAPMODE);FETCH PREVIOUS CC'S & EXT BIT 6680.000
                                                *            AND SHIFT VALID & PROTECT TO 05/06 AND 21/22              6681.000
0CD5 0 0 0 0 1 E E 0 02 2 50C 0 0 0 0 0 0 0 0          FULLMAR=@50000000, TNIBL, SETXCC(S); SHIFT VALID & PROTECT TO   6682.000
                                                *            01/02 AND 17/18.                                          6683.000
0CD6 0 6 0 4 0 A 9 4 00 0 829 0 0 0 0 0 0 0 0  829     R(R)=R(R):T, *GO TO FETCH.RETURN; COMBINE VALID & PROTECT WITH  6684.000
                                                *                 MAP ENTRIES.                                         6685.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                          M A P   C O N T E X T   S W I T C H
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                        6686.000
                                                        **                                                              6687.000
                                                        **      READ MEMORY FOR POWER ON RESTART                        6688.000
                                                        **                                                              6689.000
                                                        (@CD7)                                                          6690.000
                                                        IPU.POWER.ON                                                    6691.000
0CD7 4 6 0 0 1 0 0 0 00 2 F3F 0 0 0 0 0 0 0 0  F3E              IF IPU *GO TO POWER.UP.S.P.CLEAR; SET UP PANEL ETC.     6692.000
0CD8 0 0 0 0 1 0 0 0 1C 1 080 0 0 0 0 0 0 0 0                   READ, FRCWORD;   GET SAVED SCRATCH PAD INFOMATION       6693.000
0CD9 9 4 0 0 1 0 0 0 00 0 75B 0 0 0 0 0 0 0 0  CDB              IF %OPRNDPE:OPNORESP *GO TO $+2;                       6694.000
0CDA 0 6 0 0 0 1 F 0 00 F F3E 0 0 0 0 0 0 0 0  F3E            T=%T(ZE), *GO TO POWER.UP.ERROR;  SET PWR UP ERROR FLAGS  6695.000
0CDB 0 6 0 0 1 0 0 0 00 0 10C 0 0 0 0 0 0 0 0  10C              *GO TO POWER.READ.OK; DATA READ FROM MEMORY IS OK       6696.000
                                                        *                                                               6697.000
                                                        *       IPU POWER DOWN EXIT                                     6698.000
                                                        *                                                               6699.000
                                                        IPU.POWER.DWN                                                   6700.000
0CDC 0 0 0 0 4 0 2 0 00 0 E00 0 0 0 0 0 0 0 0                   PC=R(TEMP4); RESTORE PC FROM BEFORE PF                  6701.000
0CDD 0 0 0 2 1 E F 0 02 0 020 0 0 0 0 0 0 0 0                   T=@00000200; SET IPU POWER FAIL FLAG BIT 22             6702.000
0CDE 0 6 0 0 1 0 0 0 00 0 2D4 0 0 0 0 0 0 0 0  2D4              *GO TO IPU.ERROR.TRAP; TRAP POWER FAIL ERROR            6703.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FLOATING POINT PRIMARY DECODE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6704.000

                                                                                                                    6705.000
                                                *** PRIMARY DECODE FOR FLOATING POINT INSTRUCTIONS ***               6706.000
                                                *** OP CODES E0 AND E4  ; 0 = 38 AND 39 ***                          6707.000
                                                *** NOTE :  THIS DECODE SEPARATES HARDWARE AND FIRMWARE FLOATING     6708.000
                                                ***         POINT OPERATIONS BY THE STATE OF THE LENFP SIGNAL TO     6709.000
                                                ***         THE DECODE LOGIC.                                        6710.000
                                                ***                                                                  6711.000
                                                                                                                    6712.000
                                                                                                                    6713.000

                                                (@CE0)                                                              6714.000

                                                ADD.SUB.FLT.PT                                                      6715.000

                                                MPY.DVD.FLT.PT                                                      6716.000

                                                        MAPIX=R(X)+I0, RSTRHF, DECODE(#), RESET(FPDWORD),           6717.000

0CE0 5 2 6 4 2 3 7 2 0C C 606 0 0 0 0 0 0 0 0           IF %LATERRW *JUMPD;INITIALIZE HARDWARE OR FIRMWARE FLOATING PT 6718.000

0CE1 0 6 0 0 1 0 0 0 00 0 460 0 0 0 0 0 0 0 0  460      *GO TO LEXTPROC;  EXIT TO LATE ERROR PROCESSING             6719.000

                                                                                                                    6720.000
                                                *** HARDWARE FLOATING POINT INDIRECT SCHEDULING ***                  6721.000
                                                *** FP.INDIRECT                                                     6722.000
                                                                                                                    6723.000
                                                                                                                    6724.000

                                                FP.INDIRECT                                                         6725.000

0CE2 0 0 5 0 3 0 0 0 00 0 050 0 0 0 0 0 0 0 0           NOD=DI, DECODE(@15);  SET SINGLE PRECISION DECODE            6726.000

0CE3 0 0 0 4 3 3 7 1 00 0 000 0 0 0 0 0 0 0 0           MAPIX=R(DIX)+DI;  USE INDIRECT INDEX PATH                    6727.000

0CE4 2 4 0 0 1 0 0 0 1C 1 18E 0 0 0 0 0 0 0 0  CEE      READ, FRCWORD, IF EXTL *HOP FP.CURRENT.EXIT;                6728.000

0CE5 2 4 0 0 3 0 0 0 09 0 C02 0 0 0 0 0 0 0 0  CE2      NOD=DI, RSTFP, IF INDIR *GO TO FP.INDIRECT;                 6729.000

0CE6 0 0 0 0 3 0 0 0 09 0 000 0 0 0 0 0 0 0 0           RSTFP, NOD=DI;   CLEAR FLOATING POINT HARDWARE              6730.000

0CE7 A 2 0 0 1 0 0 0 00 0 E00 0 0 0 0 0 0 0 0           IF %FC1V *JUMPD;   EXIT IF SINGLE PRECISION                 6731.000

0CE8 0 0 5 0 1 0 0 0 00 0 060 0 0 0 0 0 0 0 0           DECODE(@16);   SET DOUBLE PRECISION DECODE                  6732.000

0CE9 0 2 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPD;  EXIT TO DOUBLE PRECISION                           6733.000

                                                *                                                                   6734.000

                                                (@CEA)                                                              6735.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
FLOATING POINT PRIMARY DECODE

```
SEL 32/75 CPU
PC  TSMAB+DR YX PCH+BDM+BAD ADDR

                                                                                                                    6737.000
                                                    *                                                               6738.000
                                                    IPU.BRI                                                         6739.000
                                                    *                                                               6740.000
0CEA 4 6 0 0 1 0 0 0 00 2 7F5 0 0 0 0 0 0 0 0  7F5      IF IPU *GOTO UNDEF.75;
0CEB 0 0 0 1 6 D 1 0 02 0 7F0 0 0 0 0 0 0 0 0            S=@007F0000&INTLVL;       GET CURRENT POLLING LEVEL     6741.000
0CEC 0 6 0 0 1 0 0 0 00 0 41D 0 0 0 0 0 0 0 0  41D      *GOTO BRI.RTN;                                          6742.000
                                                    (@CED)                                                          6743.000
                                                    FP.DOUBLE.CHECK                                                 6744.000
0CED 9 1 0 0 1 0 0 0 08 0 550 0 0 0 0 0 0 0 0            RDLSW, IF %OPRNDPF:OPTIMEOUT:OPNORESP *JUMPJ;           6745.000
                                                    FP.CURRENT.EXIT                                                 6746.000
0CEE 0 6 0 0 1 0 0 0 00 0 45D 0 0 0 0 0 0 0 0  45D      *GO TO EXTPROC1;   EXIT TO CURRENT ERROR INSTRUCION     6747.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP ADD SINGLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    6748.000
                                                    ****** ADDITON FLOATING POINT WORD SINGLE PRECISION ******      6749.000
                                                    (@CF0)                                                          6750.000
                                                    FFPADDS                                                         6751.000
0CF0 0 0 0 0 0 0 4 1C 1 080 0 0 0 3 0 0 0 0         RD=R(R) CPAOB, READ, FRCWORD; READ MEMORY AND LOAD RD           6752.000
0CF1 D 0 5 2 4 7 0 0 00 0 4D0 3 3 1 0 0 0 2 2       RA=SHIFT(RD(SLA,@07)),DSFPW,RAE=EXPK+RDEXP,DECODE(@1D);         6753.000
0CF2 2 5 0 0 1 0 0 0 00 0 CE2 0 0 0 0 0 0 0 0 CE2   IF INDIR *GO TO FP.INDIRECT;  EXIT IF INDIRECT FP OPERATION    6754.000
0CF3 5 6 0 0 1 0 0 0 00 7 D88 0 0 0 0 0 0 0 0 D88   IF %LFC1V *GO TO CHANGE.TO.DOUBLE;                           6755.000
0CF4 8 0 0 0 3 0 0 0 00 0 003 3 3 2 0 0 0 0 0       DFLSHF, RB=SHIFT(RD(DFLSHFL));                                  6756.000
0CF5 F 7 3 0 1 0 0 0 00 0 000 1 0 0 0 0 0 0 0       *JUMPAV, NOD=RA+RB; VECTORED JUMP TO JAV1 PLUS 1, 3, OR 13      6757.000
                                                    JAV1                                                            6758.000
0CF6 F 7 1 0 1 0 0 0 00 0 000 1 0 0 0 0 0 3 2       NOD=RA+RB,RAE=RMEXP, *JUMPNV;                                   6759.000
                                                    JAV1.01(JAV1+1)                                                 6760.000
                                                    DEXP.GT.SIX.ADD.SING                                            6761.000
0CF7 D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0       NOR2, NOD=SHIFT(RA(SLA,@00));                                   6762.000
0CF8 0 4 0 0 1 0 0 0 19 0 08E 0 0 0 0 0 0 0 0 CFE   FETCHPC, *GO TO JNV1.05;                                         6763.000
                                                    JAV1.03(JAV1+3)                                                 6764.000
                                                    DEXP.ZERO.ADD.SING                                              6765.000
                                                    JNV1                                                            6766.000
0CF9 D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 1 2 2       RND, NOR2, NOD=SHIFT(RA(SLA, 00)), RAE=EXPK+RAE, FETCHPC;       6767.000
                                                    ***  VECTORED JUMP FOR NORMALIZE (JUMP TO JNV1 PLUS 1, 3, OR 5) 6768.000
                                                    JNV1.01(JNV1+1)                                                 6769.000
                                                    R.RND.ADD.SING.1                                                6770.000
0CFA D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0       R(R)=SHIFT(RP(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#), *JUMPZ;   6771.000
                                                    JNV1.03(JNV1+3)                                                 6772.000
                                                    R.ZERO.ADD.SING.1                                               6773.000
0CFC 0 3 1 2 4 0 9 0 00 0 000 3 1 4 0 0 0 0 0       R(R)=SHIFT(RA(SLA,00)),SETCC(#), *JUMPZ; RESULT = 0             6774.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 32/75 C P U F P A D D S I N G L E
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                        JNV1.05(JNV1+5)                                                    6775.000
                                        R.NOR.ADD.SING.1                                                   6776.000
0CFE 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2       RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; NOT NORMALIZED   6777.000
0CFF 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2       CORRMNG, RAE=FXPK+RAE;                                     6778.000
0D00 D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0       R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#), *JUMPZ; 6779.000
                                        JAV1.13(JAV1+13)                                                   6780.000
                                        DEXP.LT.SIX.ADD.SING                                               6781.000
0D03 F 7 1 0 5 0 0 0 00 0 000 3 1 0 0 0 1 2 2       *JUMPNV,RAE=EXPK+RAE, NOD=SHIFT(RA(SLA,HWDCNT));           6782.000
                                        ***  VECTORED JUMP FOR NORMALIZE (JUMP TO JNV2 PLUS 1, 3, OR 5)    6783.000
                                        JNV2                                                               6784.000
0D04 D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 0 0 0       RND,NOR2,NOD=SHIFT(RA(SLA,00)), FETCHPC;                   6785.000
                                        JNV2.01(JNV2+1)                                                    6786.000
                                        R.RND.ADD.SING.2                                                   6787.000
0D05 D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0       R(R)=SHIFT(RP(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#), *JUMPZ; 6788.000
                                        JNV2.03(JNV2+3)                                                    6789.000
                                        R.ZERO.ADD.SING.2                                                  6790.000
0D07 0 3 1 2 4 0 9 0 00 0 000 3 1 4 0 0 0 0 0       R(R)=SHIFT(RA(SLA,00)),SETCC(#), *JUMPZ; RESULT = 0        6791.000
                                        JNV2.05(JNV2+5)                                                    6792.000
                                        R.NOR.ADD.SING.2                                                   6793.000
0D09 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2       RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT;RESULT NOT NORMALIZED 6794.000
0D0A 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2       CORRMNG, RAE=FXPK+RAE;                                     6795.000
0D0B D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0       R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#), *JUMPZ; 6796.000
*                                                                                                          6797.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP ADD SINGLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                    6798.000
                                        (@D0C)                                      6799.000
                                        *                                           6800.000
                                        *              TEST FOR IPU WAIT            6801.000
                                        *                                           6802.000
                                        IPU.WAIT                                    6803.000
                                        *                                           6804.000
0D0C 9 4 0 0 1 0 0 0 00 0 B8E 0 0 0 0 0 0 0 0  D0E      IF %PRIVBIT *GOTO $+2;     6805.000
0D0D 4 6 0 0 1 0 0 0 00 2 9EC 0 0 0 0 0 0 0 0  9FC      IF IPU *GOTO PRIV.ERROR5; REPORT PRIV VIOLATION   6806.000
0D0E 0 0 0 2 1 E 1 0 02 0 E04 0 0 0 0 0 0 0 0            SET(DWAIT),S=@0000E000;  SET 34.5 MS COUNT IN S REG   6807.000
0D0F 0 6 0 0 1 0 0 0 00 0 9F6 0 0 0 0 0 0 0 0  9F6      *GOTO WAIT.RTN;            6808.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP SUBTRACT SINGLE
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                6809.000
                                                                    ******  SUBTRACTION FLOATING POINT WORD SINGLE PRECISION  ******    6810.000
                                                                    ($+15&-16)                                                      6811.000
                                                                    FFPSUBS                                                         6812.000
0D10 0 0 0 0 0 0 0 4 1C 1 080 0 0 0 3 0 0 0 0                               RD=R(R) CPAOB, READ, FRCWORD; READ MEMORY AND LOAD RD      6813.000
0D11 D 0 5 2 4 7 0 0 00 0 4D0 3 3 1 0 0 0 2 2                               RA=SHIFT(RD(SLA,@07)),DSFPW,RAE=EXPK+RDEXP,DECODE(@1D);     6814.000
0D12 2 6 0 0 1 0 0 0 00 0 CE2 0 0 0 0 0 0 0 0  CE2                          IF INDIR *GO TO FP.INDIRECT;  EXIT IF INDIRECT FP OPERATION 6815.000
0D13 2 5 0 0 1 0 0 0 00 0 E88 0 0 0 0 0 0 0 0  D88                          IF FC1V *GO TO CHANGE.TO.DOUBLE;                            6816.000
0D14 8 0 0 0 3 0 0 0 00 0 003 3 3 2 0 0 0 0 0                               DFLSHF, RB=SHIFT(RD(DFLSHFL));                              6817.000
0D15 F 7 3 0 1 0 0 0 00 0 000 2 0 0 0 0 0 0 0                               *JUMPAV, NOD=RA-RB;  VECTORED JUMP TO JAV3 PLUS 1, 3, OR 13  6818.000
                                                                    JAV3                                                            6819.000
0D16 F 7 1 0 1 0 0 0 00 0 000 2 0 0 0 0 0 3 2                               NOD=RA-RB, RAE=RMEXP, *JUMPNV;                              6820.000
                                                                    JAV3.01(JAV3+1)                                                 6821.000
                                                                    DEXP.GT.SIX.SUB.SING                                            6822.000
0D17 D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0                               NOR2, NOD=SHIFT(RA(SLA,@00));                               6823.000
0D18 0 4 0 0 1 0 0 0 19 0 08F 0 0 0 0 0 0 0 0  D1E                          FETCHPC, *GO TO JNV3.05;                                    6824.000
                                                                    JAV3.03(JAV3+3                                                  6825.000
                                                                    DEXP.ZERO.SUB.SING                                              6826.000
                                                                    JNV3                                                            6827.000
0D19 D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 1 2 2                               RND, NOR2, NOD=SHIFT(RA(SLA,00)), RAE=EXPK+RAE, FETCHPC;    6828.000
                                                                    ***  VECTORED JUMP FOR NORMALIZE (JUMP TO JNV3 PLUS 1, 3, OR 5)  6829.000
                                                                    JNV3.01(JNV3+1)                                                 6830.000
                                                                    R.RND.SUB.SING.1                                                6831.000
0D1A D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0                               R(R)=SHIFT(RP(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMP7; 6832.000
                                                                    JNV3.03(JNV3+3)                                                 6833.000
                                                                    R.ZERO.SUB.SING.1                                               6834.000
0D1C 0 3 1 2 4 0 9 0 00 0 000 3 1 4 0 0 0 0 0                               R(R)=SHIFT(RA(SLA,00)),SETCC(#),*JUMP7; RESULT = 0          6835.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP SUBTRACT SINGLE

PC	TSMAB+DR YX PCH+BDM+BAD ADDR		
		JNV3.05(JNV3+5)	6836.000
		R.NOR.SUB.SING.1	6837.000
0D1E	0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2	RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; NOT NORMALIZED	6838.000
0D1F	8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2	CORRMNG, RAE=EXPK+RAE;	6839.000
0D20	D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0	R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ;	6840.000
		JAV3.13(JAV3+13)	6841.000
		DEXP.LT.SIX.SUB.SING	6842.000
0D23	F 7 1 0 5 0 0 0 00 0 000 3 1 0 0 0 1 2 2	*JUMPNV,RAE=EXPK+RAE, NOD=SHIFT(RA(SLA,HWDCNT));	6843.000
		*** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV4 PLUS 1, 3, OR 5)	6844.000
		JNV4	6845.000
0D24	D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 0 0 0	RND,NOR2,NOD=SHIFT(RA(SLA,00)), FETCHPC;	6846.000
		JNV4.01(JNV4+1)	6847.000
		R.RND.SUB.SING.2	6848.000
0D25	D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0	R(R)=SHIFT(RP(SPA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ;	6849.000
		JNV4.03(JNV4+3)	6850.000
		R.ZERO.SUB.SING.2	6851.000
0D27	0 3 1 2 4 0 9 0 00 0 000 3 1 4 0 0 0 0 0	R(R)=SHIFT(RA(SLA,00)),SETCC(#),*JUMPZ; RESULT = 0	6852.000
		JNV4.05(JNV4+5)	6853.000
		R.NOR.SUB.SING.2	6854.000
0D29	0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2	RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; RESULT %NORMALIZED	6855.000
0D2A	8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2	CORRMNG, RAE=EXPK+RAE;	6856.000
0D2B	D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0	R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ;	6857.000
		*	6858.000

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
FP SUBTRACT SINGLE

```
SEL 32/75 CPU
PC  TSMAB+DR YX PCH+BDM+BAD ADDR
                                                                                                                    6859.000
                                                                                                                    6860.000
                                                *                IPU TEST FOR CONTROL INSTRUCTION                   6861.000
                                                *                                                                   6862.000
                                                (@D2C)                                                              6863.000
                                                *                                                                   6864.000
                                                IPU.CTL                                                             6865.000
                                                *                                                                   6866.000
0D2C 4 6 0 0 1 0 0 0 00 2 7F5 0 0 0 0 0 0 0 0  7F5      IF IPU *GOTO UNDEF.75;                                     6867.000
0D2D 5 4 0 4 2 D 1 0 02 C 03F 0 0 0 0 0 0 0 0  D2F      S=@03FFFFFF&IO,IF %LATERRW *GOTO $+2;                      6868.000
0D2E 0 6 0 0 1 0 0 0 0C 0 460 0 0 0 0 0 0 0 0  460      &STRHF,*GOTO LEXTPROC;                                     6869.000
0D2F 0 6 0 0 1 0 0 0 00 0 802 0 0 0 0 0 0 0 0  802      *GOTO INTR.CTL.RTN;                                        6870.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP ADD DOUBLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6671.000
                                                    ******* ADD FLOATING POINT WORD DOUBLE   ********               6872.000
                                                    ($+158-16)                                                      6873.000
                                                    FPPADDD                                                         6874.000
0D30 0 0 0 0 0 0 0 6 1C 1 F86 0 0 0 3 0 0 0 0           RD=R(RD) CPADR, READ, FRCWORD, SET(FPDWORD);                6875.000
0D31 0 0 5 0 0 0 0 4 00 0 0D0 0 0 0 1 0 0 0 0           RD=R(R) CPSOB,DFCODE(@1D); REGISTER OPERAND                 6876.000
0D32 D 0 0 2 4 7 0 0 00 0 400 3 3 1 0 0 0 2 2           RA=SHIFT(RD(SLA,@07)),DSFPW,RAE=EXPK+RDEXP;                 6877.000
0D33 2 6 0 0 1 0 3 0 00 0 CE2 0 0 0 0 0 0 0 0  CE2      MAR=MAR, IF INDIR *GO TO FP.INDIRECT;  EXIT IF INDIRECT      6878.000
0D34 0 0 0 0 1 0 0 0 1C 0 082 0 0 0 0 0 0 0 0           READ,FORCE2; READ LSWRD                                     6879.000
0D35 A 5 0 0 3 0 0 0 00 0 E8C 0 0 0 0 0 0 0 0  D8C      NOD=DI, IF %FC1V *GO TO CHANGE.TO.SINGLE;                   6880.000
0D36 0 6 7 0 1 0 0 0 08 0 CED 0 0 0 0 0 0 0 0  CED      RDLSW, *LINK FP.DOUBLE.CHECK;                               6881.000
0D37 8 0 0 0 3 0 0 0 08 0 003 3 3 2 0 0 0 0 0           RDLSW, DELSHF, RB=SHIFT(RD(DELSHFL));                       6882.000
0D38 F 7 3 0 1 0 0 0 00 0 000 1 0 0 0 0 0 0 0           *JUMPAV, NOD=RA+RB;  VECTORED JUMP TO JAV5 PLUS 1, 3, OR 13 6883.000
                                                    JAV5                                                            6884.000
0D39 F 7 1 0 1 0 0 0 00 0 000 1 0 0 0 0 0 3 2           NOD=RA+RB,RAE=RMEXP, *JUMPNV;                               6885.000
                                                    JAV5.01(JAV5+1)                                                 6886.000
                                                    DEXP.GT.13.ADD.DOUBLE                                           6887.000
0D3A D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0           NOR2, NOD=SHIFT(RA(SLA,@00));                               6888.000
0D3B 0 5 0 0 1 0 0 0 00 0 041 0 0 0 0 0 0 0 0  D41      *GO TO JNV5.05;                                             6889.000
                                                    JAV5.03(JAV5+3)                                                 6890.000
                                                    DEXP.ZERO.ADD.DOUBLE                                            6891.000
                                                    JNV5                                                            6892.000
0D3C D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 1 2 2           NOR2, NOD=SHIFT(RA(SLA,00)), RAE=EXPK+RAE;                  6893.000
                                                    *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV5 PLUS 1, 3, OR 5)  6894.000
                                                    JNV5.01(JNV5+1)                                                 6895.000
                                                    R.RND.ADD.DOUBLE.1                                              6896.000
0D3D D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC; 6897.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP ADD DOUBLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0D3E 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                          6898.000
                                                JNV5.03(JNV5+3)                                                6899.000
                                                R.ZERO.ADD.DOUBLE.1                                            6900.000
0D3F 0 0 1 2 4 0 9 0 19 0 080 3 1 4 0 0 0 0 0        R(R)=SHIFT(RA(SLA,00)),SETCC(#), FETCHPC;                 6901.000
0D40 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                          6902.000
                                                JNV5.05(JNV5+5)                                                6903.000
                                                R.NOR.ADD.DOUBLE.1                                             6904.000
0D41 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2        RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-RRE; NOT NORMALIZED     6905.000
0D42 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2        CORRMNG, RAE=EXPK+RAE;                                    6906.000
0D43 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0        R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC; 6907.000
0D44 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                          6908.000
                                                JAV5.13(JAV5+13)                                               6909.000
                                                DEXP.LT.13.ADD.DOUBLE                                          6910.000
0D46 F 7 1 0 1 0 0 0 00 0 000 0 0 0 0 0 1 2 2        RAE=EXPK+RAE, *JUMPNV;  VECTORED JUMP TO JNV6 PLUS 1, 3, OR 5 6911.000
                                                JNV6                                                           6912.000
0D47 D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0        NOR2,NOD=SHIFT(RA(SLA,00));                               6913.000
                                                JNV6.01(JNV6+1)                                                6914.000
                                                R.RND.ADD.DOUBLE.2                                             6915.000
0D48 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0        R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC; 6916.000
0D49 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                          6917.000
                                                JNV6.03(JNV6+3)                                                6918.000
                                                R.ZERO.ADD.DOUBLE.2                                            6919.000
0D4A 0 0 1 2 4 0 9 0 19 0 080 3 1 4 0 0 0 0 0        R(R)=SHIFT(RA(SLA,00)),SETCC(#), FETCHPC;                 6920.000
0D4B 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                          6921.000
                                                JNV6.05(JNV6+5)                                                6922.000
                                                R.NOR.ADD.DOUBLE.2                                             6923.000
0D4C 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2        RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-RRE; RESULT NOT NORMALIZED 6924.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
S E L 3 2 / 7 5 C P U F P A D D D O U B L E

```
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR

0D4D 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2        CORRMNG, RAF=EXPK+RAE;                                                    6925.000

0D4E D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0        R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;             6926.000

0D4F 0 3 0 3 5 9 8 6 00 0 000 3 1 4 0 0 0 0 0        R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                          6927.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    F P  S U B T R A C T  D O U B L E
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    6928.000
                                                    *******   SUBTRACT FLOATING POINT WORD DOUBLE   ********        6929.000
                                                    ($+15&-16)                                                      6930.000
                                                    FFPSUBD                                                         6931.000
0D50 0 0 0 0 0 0 0 6 1C 1 F86 0 0 0 3 0 0 0 0           RD=R(RO) CPAOB, READ, FRCWORD, SET(FPDWORD);                6932.000
0D51 0 0 5 0 0 0 0 4 00 0 0D0 0 0 0 1 0 0 0 0           RD=R(R) CPSOB,DECODF(@1D); REGISTER OPERAND                 6933.000
0D52 0 0 0 2 4 7 0 0 00 0 400 3 3 1 0 0 0 2 2           RA=SHIFT(RD(SLA,@07)),DSFPW,RAE=EXPK+RDEXP;                 6934.000
0D53 2 6 0 0 1 0 3 0 00 0 CE2 0 0 0 0 0 0 0 0  CE2      MAR=MAR, IF INDIR *GO TO FP.INDIRECT;  EXIT IF INDIRECT     6935.000
0D54 0 0 0 0 1 0 0 0 1C 0 082 0 0 0 0 0 0 0 0           READ,FORCFZ; READ LSWRD                                     6936.000
0D55 A 5 0 0 3 0 0 0 00 0 E8C 0 0 0 0 0 0 0 0  D8C      NOD=DI, IF %FC1V *GO TO CHANGE.TO.SINGLE;                   6937.000
0D56 0 6 7 0 1 0 0 0 08 0 CED 0 0 0 0 0 0 0 0  CED      RDLSW, *LINK FP.DOUBLE.CHECK;                               6938.000
0D57 8 0 0 0 3 0 0 0 08 0 003 3 3 2 0 0 0 0 0           RDLSW, DELSHF, RB=SHIFT(RD(DELSHFL));                       6939.000
0D58 F 7 3 0 1 0 0 0 00 0 000 2 0 0 0 0 0 0 0           *JUMPAV, NOD=RA-RB;  VECTORED JUMP TO JAV7 PLUS 1, 3, OR 13 6940.000
                                                    JAV7                                                            6941.000
0D59 F 7 1 0 1 0 0 0 00 0 000 2 0 0 0 0 0 3 2           NOD=RA-RB,RAE=RMEXP, *JUMPNV;                               6942.000
                                                    JAV7.01(JAV7+1)                                                 6943.000
                                                    DEXP.GT.SIX.SUB.DOUBLE                                          6944.000
0D5A D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0           NOR2, NOD=SHIFT(RA(SLA,@00));                               6945.000
0D5B 0 5 0 0 1 0 0 0 00 0 061 0 0 0 0 0 0 0 0  D61      *GO TO JNV7.05;                                             6946.000
                                                    JAV7.03(JAV7+3)                                                 6947.000
                                                    DEXP.ZERO.SUB.DOUBLE                                            6948.000
                                                    JNV7                                                            6949.000
0D5C D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 1 2 2           NOR2, NOD=SHIFT(RA(SLA,00)), RAE=EXPK+RAE;                  6950.000
                                                    *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV7 PLUS 1, 3, OR 5)  6951.000
                                                    JNV7.01(JNV7+1)                                                 6952.000
                                                    R.RND.SUB.DOUBLE.1                                              6953.000
0D5D D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC; 6954.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  CPU                                                                    F P   S U B T R A C T   D O U B L E
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0D5E 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0          R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                    6955.000
                                                    JNV7.03(JNV7+3)                                                        6956.000
                                                    R.ZERO.SUB.DOUBLE.1                                                    6957.000
0D5F 0 0 1 2 4 0 9 0 19 0 080 3 1 4 0 0 0 0 0          R(R)=SHIFT(RA(SLA,00)),SETCC(#), FETCHPC;                           6958.000
0D60 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0          R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                    6959.000
                                                    JNV7.05(JNV7+5)                                                        6960.000
                                                    R.NOR.SUB.DOUBLE.1                                                     6961.000
0D61 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2          RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; NOT NORMALIZED            6962.000
0D62 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2          CORRMNG, RAF=FXPK+RAE;                                              6963.000
0D63 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0          R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;       6964.000
0D64 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0          R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                    6965.000
                                                    JAV7.13(JAV7+13)                                                       6966.000
                                                    DEXP.LT.SIX.SUB.DOUBLE                                                 6967.000
0D66 F 7 1 0 1 0 0 0 00 0 000 0 0 0 0 0 1 2 2          RAE=EXPK+RAE, *JUMPNV;                                              6968.000
                                                    *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV8 PLUS 1, 3, OR 5)         6969.000
                                                    JNV8                                                                   6970.000
0D67 D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0          NOR2,NOD=SHIFT(RA(SLA,00));                                         6971.000
                                                    JNV8.01(JNV8+1)                                                        6972.000
                                                    R.RND.SUB.DOUBLE.2                                                     6973.000
0D68 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0          R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;       6974.000
0D69 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0          R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                    6975.000
                                                    JNV8.03(JNV8+3)                                                        6976.000
                                                    R.ZERO.SUB.DOUBLE.2                                                    6977.000
0D6A 0 0 1 2 4 0 9 0 19 0 080 3 1 4 0 0 0 0 0          R(R)=SHIFT(RA(SLA,00)),SETCC(#), FETCHPC;                           6978.000
0D6B 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0          R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                    6979.000
                                                    JNV8.05(JNV8+5)                                                        6980.000
                                                    R.NOR.SUB.DOUBLE.2                                                     6981.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP SUBTRACT DOUBLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
0D6C 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2        RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; RESULT %NORMALIZED   6982.000
0D6D 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2        CORRMNG, RAE=FXPK+RAE;                                         6983.000
0D6E D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0        R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=FXPK+RAE,SFTCC(#),FETCHPC;  6984.000
0D6F 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0        R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                               6985.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP MULTIPLY SINGLE
PC T S M A B + D R Y X PCH + B D M + B A D ADDR

```
                                                                                                                                6986.000
                                                                    ******  SINGLE PRECISION MULTIPLY  ******                    6987.000
                                                                    FFPMPYS                                                     6988.000
0D70 0 0 0 0 0 0 0 4 1C 1 080 0 0 0 3 0 0 0 0                               RD=R(R) CPAOB, PEAD, FRCWORD; READ MEMORY AND LOAD RD       6989.000
0D71 0 0 5 2 4 7 0 0 00 0 4F0 3 3 2 0 0 0 2 1                               RB=SHIFT(RD(SLA,@07)),DSFPW,RBE=EXPK+RDEXP,DECODE(@1F);     6990.000
0D72 0 0 0 0 1 0 0 0 00 0 205 2 0 1 0 0 0 2 1                               COMPLAN,RA=RA-RB,CORRZEROW,RBE=EXPK+RBE;                   6991.000
0D73 2 6 0 0 1 0 0 0 00 0 CE2 0 0 2 0 0 0 0 0  CE2                          IF INDIR *GO TO FP.INDIRECT, RB=0; EXIT IF INDIR. OPER.    6992.000
0D74 3 4 0 0 3 E C 0 02 7 056 0 0 0 0 0 0 0 0  D76                          NU=@05000000,BMUX=DT,IF LFC1V *HOP $+2;SET MPY ITERATION CNT 6993.000
0D75 0 5 0 0 1 0 0 0 00 0 088 0 0 0 0 0 0 0 0  D88                          *GO TO CHANGE.TO.DOUBLE; EXIT IF DOUBLE WORD C-BITS         6994.000
0D76 0 0 0 2 4 0 0 0 00 0 000 3 3 2 5 0 0 3 2                               RB=SHIFT(RD(SLA,00)),RAE=RMEXP,MPY(D);XFR MULTIPLICAND TO RB 6995.000
0D77 0 0 0 0 1 0 0 0 00 0 205 2 0 2 0 0 1 2 2                               RB=RA-RB,COMPLAN, RAE=EXPK+RAE, CORRZEROW; COMP B IF <0     6996.000
0D78 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                               *NOP;                                                      6997.000
0D79 0 0 0 2 1 9 0 0 00 0 000 3 1 6 0 0 0 0 2                               RD=SHIFT(RA(SRA,-@27)), RAE=RAE+RBE;LOAD MULTIPLIER IN RD  6998.000
0D7A 0 0 0 2 3 C 0 0 00 0 000 0 0 1 6 0 0 0 0                               SPSHIFT(RD(SRA,-@04)), RA=0, MPY(D); MPY FIRST CYCLE       6999.000
                                                                            SPSHIFT(RD(SRA,-@04)),RA=RA+RP,MPY(D),MPYLOOP,DECRN,       7000.000
0D7B A 4 3 2 3 C 0 0 00 0 90B 1 2 1 6 0 0 1 0  D7B                          IF %NCTRZ *GO TO $;    MULTIPLY                            7001.000
0D7C 0 0 0 0 1 0 0 0 00 0 000 1 2 2 5 0 0 0 0                               RB=RA+RP,MPY(D); MPY LAST CYCLE                            7002.000
0D7D 8 0 0 0 1 0 0 0 00 0 007 2 0 5 0 0 0 0 0                               COMPLAF, RDMS=RA-RB; COMPLEMENT IF SIGN F/F IS SET          7003.000
0D7E 0 0 0 2 4 0 0 0 00 0 000 3 3 1 0 0 0 0 0                               RA=SHIFT(RD(SLA,0)); LOAD MOST SIGNIFICANT WORD            7004.000
0D7F F 7 1 2 4 0 0 0 00 0 600 3 1 1 0 0 0 0 0                               *JUMPNV, RA=SHIFT(RA(SLA,00)), MASK;                       7005.000
                                                                    *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV9 PLUS 1, 3, OR 5)  7006.000
                                                                    JNV9                                                        7007.000
0D80 D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 0 0 0                               RND,NOR2,NOD=SHIFT(RA(SLA,00)), FETCHPC;                   7008.000
                                                                    JNV9.01(JNV9+1)                                             7009.000
                                                                    R.RND.MULTIPLY.SING.1                                       7010.000
0D81 D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0                               R(R)=SHIFT(RP(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ ; 7011.000
                                                                    JNV9.03(JNV9+3)                                             7012.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          F P  M U L T I P L Y  S I N G L E
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                    R.ZERO.MULTIPLY.SING.1                                                  7013.000

0D83 0 3 1 2 4 0 9 0 00 0 000 3 1 4 0 0 0 0 0           R(R)=SHIFT(RA(SLA,00)),SETCC(#),*JUMPZ ; RESULT = 0                     7014.000

                                                    JNV9.05(JNV9+5)                                                         7015.000

                                                    R.NOR.MULTIPLY.SING.1                                                   7016.000

0D85 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2           RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; RESULT %NORMALIZED            7017.000

0D86 8 0 0 2 4 0 0 0 00 0 002 3 1 0 0 0 1 2 2           CORRMNG, RAE=EXPK+RAE, NOD=SHIFT(RA(SLA,@00));                          7018.000

0D87 D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ ;           7019.000

                                                                                                                            7020.000
                                                    ***                                                                     7021.000
                                                    *** CHANGE FROM SINGLE TO DOUBLE PRECISION DUE TO INDEXING ***          7022.000
                                                    ***                                                                     7023.000

                                                                                                                            7024.000
                                                    CHANGE.TO.DOUBLE                                                        7025.000

0D88 0 0 5 0 3 0 0 0 09 0 060 0 0 0 0 0 0 0 0           NOD=DI, RSTFP, DECODE(@16); SET DOUBLE PRECISION DECODE                 7026.000

0D89 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                                   7027.000

0D8A 0 2 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPD;  JUMP TO DOUBLE PRECISION INSTRUCTIONS                          7028.000

0D8B 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                                   7029.000

                                                                                                                            7030.000
                                                    ***                                                                     7031.000
                                                    *** CHANGE FROM DOUBLE TO SINGLE PRECISION DUE TO INDEXING ***          7032.000
                                                    ***                                                                     7033.000

                                                                                                                            7034.000
                                                    CHANGE.TO.SINGLE                                                        7035.000

0D8C 0 0 5 0 3 0 0 0 09 0 050 0 0 0 0 0 0 0 0           NOD=DI, RSTFP, DECODE(@15); SET SINGLE PRECISION DECODE                 7036.000

0D8D 0 0 0 0 1 0 0 0 00 0 606 0 0 0 0 0 0 0 0           RESET(FPDWORD);                                                         7037.000

0D8E 0 2 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *JUMPD; JUMP TO SINGLE PRECISION DECODE                                 7038.000

0D8F 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0           *NOP;                                                                   7039.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP DIVIDE SINGLE
PC T S M A B+D R Y X PCH+B D M+B A D ADDR

```
                                                                                                                    7040.000
                                                    ******  SINGLE PRECISION DIVIDE  ********                    7041.000
                                                    (@D90)                                                          7043.000
                                                    FFPDVDS                                                         7044.000
0D90 0 0 0 0 0 0 0 4 1C 1 080 0 0 0 3 0 0 0 0                RD=R(R) CPA0B, READ, FRCWORD; READ MEMORY AND LOAD RD  7045.000
0D91 D 0 5 2 4 7 0 0 00 0 4F0 3 3 2 0 0 0 2 1                RB=SHIFT(RD(SLA,@07)),DSFPW,RBE=EXPK+RDEXP,DECODE(@1F);  7046.000
0D92 D 0 0 0 1 0 0 0 00 0 205 2 0 1 0 0 0 2 1                COMPLAN,RA=RA-RB,CORRZEROW,RBE=EXPK+RBE; COMPLEMENT B IF <0  7047.000
0D93 2 6 0 0 1 0 0 0 00 0 CE2 0 0 0 0 0 0 0 0 CE2            IF INDIR *GO TO FP.INDIRECT;  EXIT IF INDIRECT FP OPERATION  7048.000
0D94 3 4 0 0 3 E C 0 02 7 0E6 0 0 0 0 0 0 0 0 D96            NU=@0E000000,BMUX=DI,IF LFC1V *HOP $+2;SET DVD ITERATION CNT  7049.000
0D95 0 5 0 0 1 0 0 0 00 0 088 0 0 0 0 0 0 0 0 D88            *GO TO CHANGE.TO.DOUBLE; EXIT IF DOUBLE WORD C-BITS  7050.000
0D96 0 0 0 2 3 C 0 0 00 0 000 3 3 2 0 0 0 3 2                RB=SHIFT(RD(SRA,-@04)),RAE=RMFXP; XFR DIVISOR TO RB  7051.000
0D97 D 0 0 0 1 0 0 0 00 0 205 2 0 2 0 0 1 2 2                COMPLAN,RB=RA-RB,CORRZEROW,RAE=EXPK+RAE;COMPLEMENT B IF <0  7052.000
0D98 8 0 0 0 1 0 0 0 00 0 004 0 0 0 0 1 1 1 2                RAE=RBE-RAE, OVFMOPEZ;                                 7053.000
0D99 0 0 0 2 3 7 0 0 00 0 000 3 1 1 0 0 0 0 0                RA=SHIFT(RA(SRA,-@09));  LOAD DIVIDEND                 7054.000
0D9A F 7 2 6 5 0 0 0 00 0 000 3 0 2 0 0 0 0 0                RB=SHIFT(RB(SLA,BITCNT)), *JUMPDV;  JUMP TO JDV01 PLUS 1 OR 3  7055.000
                                                    JDV01                                                           7056.000
0D9B 0 0 0 6 4 C 0 0 0B 0 000 3 1 1 4 0 0 0 0                RA=SHIFT(RA(SLA,BITCNT)), MPY(@C), INCRN;              7057.000
                                                    JDV01.01(JDV01+1)                                               7058.000
0D9C 0 5 0 0 0 0 0 4 00 0 0AE 0 0 0 3 0 0 0 0 DAE            RD=R(R) CPA0B, *GO TO JDV01.01.EXIT; REFETCH REGISTER OPERAND  7059.000
                                                    JDV01.03(JDV01+3)                                               7060.000
0D9E 0 0 0 2 4 2 0 0 00 0 000 3 0 2 0 0 0 0 0                RB=SHIFT(RB(SLA,@02)); D/4                             7061.000
0D9F 0 0 0 0 0 2 0 0 00 0 000 0 0 0 4 0 0 0 0                MPY(@2); <RP=3D/4> , <RPP=2D/4>                        7062.000
0DA0 0 0 0 2 3 C 0 0 00 0 000 3 0 2 0 0 0 0 3                RB=SHIFT(RB(SRA,-@04)), RXE=RAE+RBE; D/4               7063.000
                                                             SPSHIFT(RD(SLL,@02)),RA=RA-RP,DIVLOOP,DECRN,IF %NCTRZ *HOP $,  7064.000
0DA1 A 4 3 3 4 2 0 0 00 0 901 2 2 0 6 0 0 2 3 DA1            RXE=EXPK+RBE;                                          7065.000
0DA2 D 0 0 2 6 2 0 0 00 0 600 3 3 2 0 0 0 0 0                RB=SHIFT(RD(SLA,@22)), MASK;                           7066.000
0DA3 F 7 1 0 1 0 0 0 00 0 007 2 0 1 0 0 0 0 0                COMPLAF,RA=RA-RB, *JUMPNV;                             7067.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP DIVIDE SINGLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV10 PLUS 1, 3, OR 5)          7068.000
                                                JNV10                                                                    7069.000
0DA4 D 0 0 2 4 0 0 0 19 0 181 3 1 0 0 0 0 0 0           RND,NOR2,NOD=SHIFT(RA(SLA,00)), FETCHPC;                         7070.000
                                                JNV10.01(JNV10+1)                                                        7071.000
0DA5 0 4 0 0 1 0 0 0 00 0 00C 0 0 0 0 0 0 0 0 DAC       *GO TO JNV10.01FXIT;                                             7072.000
                                                JNV10.03(JNV10+3)                                                        7073.000
0DA7 0 3 0 3 1 E 9 0 02 0 001 0 0 0 0 0 0 0 0           R(R)=0, SETCC(AL), *JUMPZ; SET RESULT =0 & SET CC 4 =1           7074.000
                                                JNV10.05(JNV10+5)                                                        7075.000
0DA9 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 0 0 0 0           RA=SHIFT(RA(SLA,NORMCNT)); RESULT %NORMALIZED                    7076.000
0DAA 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0           CORRMNG;                                                         7077.000
0DAB D 3 1 2 3 9 9 0 00 0 500 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ ;    7078.000
                                                JNV10.01FXIT                                                             7079.000
0DAC D 0 0 0 1 0 0 0 00 0 700 0 0 0 0 0 1 2 2           RAE=EXPK+RAE,PLUSONE;                                            7080.000
0DAD D 3 1 2 3 9 9 0 00 0 500 3 2 4 0 0 1 2 0           R(R)=SHIFT(RP(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),*JUMPZ ;    7081.000
                                                JDV01.01.EXIT                                                            7082.000
0DAE 0 0 0 2 4 0 0 0 19 0 080 3 3 1 0 0 0 0 0           RA=SHIFT(RD(SLA,0)), FETCHPC;  TEST REG OPERAND SIGN             7083.000
0DAF D 3 1 2 4 0 9 0 00 0 500 3 1 4 0 0 0 0 0           R(R)=SHIFT(RA(SLA,0)), ASFPW, SETCC(#), *JUMPZ;                  7084.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU FP MULTIPLY DOUBLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7085.000
                                                    ***     DOUBLE PRECISION MULTIPLY     ****                    7086.000
                                                    (@DB0)                                                          7087.000
                                                    FFPMPYD                                                         7088.000
0DB0 0 0 0 0 0 0 6 1C 1 E86 0 0 0 3 0 0 0 0          RD=R(RO) CPAOB, READ, FRCWORD, SET(FPDWORD);GET MOST SIGN OPR   7089.000
0DB1 2 6 0 0 0 0 0 4 00 0 CE2 0 0 0 1 0 0 0 0  CE2   RD=R(R) CPSOB, IF INDIR *GO TO FP.INDIRECT;                   7090.000
0DB2 D 0 5 2 4 7 0 0 00 0 4F0 3 3 1 0 0 0 2 1        RA=SHIFT(RD(SLA,@07)),DSFPW,RBE=EXPK+RDEXP,DECODE(@1F);         7091.000
0DB3 D 0 0 0 1 0 3 0 00 0 200 0 0 0 0 0 0 2 1        MAR=MAR, CORRZEROW, RBE=EXPK+RBE;                              7092.000
0DB4 0 0 0 0 1 0 0 0 1C 0 082 0 0 0 0 0 0 0 0        READ,FORCFZ; READ LSWRD                                        7093.000
0DB5 5 4 0 0 3 E C 0 02 7 0D7 0 0 0 0 0 0 0 0  DB7   NU=@0D000000, BMUX=DI, IF %LFC1V *HOP $+2;SET MPY ITERATION CNT 094.000
0DB6 0 5 0 0 1 0 0 0 00 0 08C 0 0 0 0 0 0 0 0  D8C   *GO TO CHANGE.TO.SINGLE; EXIT IF SINGLE PRECISION C-BITS       7095.000
0DB7 0 6 7 0 1 0 0 0 08 0 CED 0 0 2 0 0 0 0 0  CED   RDLSW, *LINK FP.DOUBLE.CHECK,RB=0;                            7096.000
0DB8 0 0 0 0 3 0 0 0 08 0 000 0 0 0 0 0 0 0 0        NOD=DI, RDLSW; WAIT FOR MEMORY OPERAND                         7097.000
0DB9 0 0 0 2 4 0 0 0 00 0 000 3 3 2 5 0 0 3 2        RB=SHIFT(RD(SLA,00)),RAE=RMEXP,MPY(D); XFR MULTIPLICAND TO RB  7098.000
0DBA D 0 0 0 1 0 0 0 00 0 200 0 0 0 0 0 1 2 2        RAE=EXPK+RAE, CORRZEROW;                                       7099.000
0DBB 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0        *NOP;                                                          7100.000
0DBC 0 0 0 2 3 9 0 0 00 0 000 3 1 6 0 0 0 0 2        RD=SHIFT(RA(SRA,-@07)), RAE=RAE+RBE;LOAD MULTIPLIER IN RD      7101.000
0DBD 0 0 0 2 3 C 0 0 00 0 000 0 0 1 6 0 0 0 0        SPSHIFT(RD(SRA,-@04)), RA=0, MPY(D); MPY FIRST CYCLE           7102.000
                                                     SPSHIFT(RD(SRA,-@04)),RA=RA+RP,MPY(D),MPYLOOP,DECRN,           7103.000
0DBE A 4 3 2 3 C 0 0 00 0 90F 1 2 1 6 0 0 1 0  DBE   IF %NCTRZ *GO TO $;    MULTIPLY                            7104.000
0DBF 0 0 0 0 1 0 0 0 00 0 000 1 2 1 5 0 0 0 0        RA=RA+RP,MPY(D); MPY LAST CYCLE                                7105.000
0DC0 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0        *NOP;                                                          7106.000
0DC1 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0        *NOP;                                                          7107.000
0DC2 8 0 0 0 1 0 0 0 00 0 006 2 0 0 0 0 0 0 0        NOD=RA-RB, COMPLAP;  MPY CORRECTION                            7108.000
0DC3 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0        *NOP;                                                          7109.000
0DC4 D 0 0 2 4 0 0 0 00 0 600 3 1 1 0 0 0 0 0        RA=SHIFT(RA(SLA,@00)), MASK;                                   7110.000
0DC5 F 7 1 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0        *JUMPNV;                                                       7111.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
FP MULTIPLY DOUBLE

```
SEL 32/75 CPU
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV11 PLUS 1, 3, OR 5)          7112.000
                                                JNV11                                                                    7113.000
0DC6 D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0           NOR2,NOD=SHIFT(RA(SLA,00));                                      7114.000
                                                JNV11.01(JNV11+1)                                                        7115.000
                                                R.RND.MULTIPLY.DOUBLE.1                                                  7116.000
0DC7 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;    7117.000
0DC8 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0           R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                 7118.000
                                                JNV11.03(JNV11+3)                                                        7119.000
                                                R.ZERO.MULTIPLY.DOUBLE.1                                                 7120.000
0DC9 0 0 1 2 4 0 9 0 19 0 080 3 1 4 0 0 0 0 0           R(R)=SHIFT(RA(SLA,00)),SETCC(#), FETCHPC;                        7121.000
0DCA 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0           R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                 7122.000
                                                JNV11.05(JNV11+5)                                                        7123.000
                                                R.NOR.MULTIPLY.DOUBLE.1                                                  7124.000
0DCB 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 1 0 0 2           RA=SHIFT(RA(SLA,NORMCNT)),RAE=RAE-NORCNT; RESULT %NORMALIZED     7125.000
0DCC 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 1 2 2           CORRMNG, RAE=EXPK+RAE;                                           7126.000
0DCD D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0           R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;    7127.000
0DCE 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0           R(RO)=SHIFT(RA(SLL,@19)),*JUMPZ;                                 7128.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU FP DIVIDE DOUBLE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7129.000
                                                    ******  DOUBLE PRECISION DIVIDE  *******                        7130.000
                                                    (@DD7)                                                          7131.000
                                                    FFPDVDD                                                         7132.000
0DD7 0 0 5 0 0 0 0 6 00 0 EF6 0 0 0 3 0 0 0 0                RD=R(RD) CPAOR, SET(FPDWORD),DECODE(@1F); GET MOST SIGN OPR 7133.000
0DD8 0 0 0 0 0 0 0 4 1C 1 080 0 0 0 1 0 0 0 0                RD=R(R) CPSOB,READ,FRCWORD;                            7134.000
0DD9 D 0 0 2 3 C 0 0 00 0 400 3 3 2 0 0 0 2 1                RB=SHIFT(RD(SRA,-@04)),DSFPW,RBE=EXPK+RDEXP;           7135.000
0DDA 2 6 0 2 4 B 0 0 00 0 CE2 3 0 2 0 0 0 0 0  CE2           RB=SHIFT(RB(SLA,@0B)), IF INDIR *GO TO FP.INDIRECT;    7136.000
0DDB D 0 0 0 1 0 3 0 00 0 205 2 0 1 0 0 0 2 1                MAR=MAR, COMPLAN, RA=RA-RB, CORRZEROW, RBE=EXPK+RBE;   7137.000
0DDC 0 0 0 0 1 0 0 0 1C 0 082 0 0 0 0 0 0 0 0                READ,FORCFZ; READ LSWRD                                7138.000
0DDD 5 4 0 0 3 E C 0 02 7 1CF 0 0 0 0 0 0 0 0  DDF           NU=@1C000000, BMUX=DI,IF %LFC1V *HOP $+2;SET DVD ITERATION CNT 7139.000
0DDE 0 5 0 0 1 0 0 0 00 0 08C 0 0 0 0 0 0 0 0  D8C           *GO TO CHANGE.TO.SINGLE; EXIT IF SINGLE PRECISION C-BITS 7140.000
0DDF 0 6 7 0 1 0 0 0 08 0 CED 0 0 0 0 0 0 0 0  CED           RDLSW, *LINK FP.DOUBLE.CHECK;                          7141.000
0DE0 0 0 0 0 3 0 0 0 08 0 000 0 0 0 0 0 0 0 0                NOD=DI, RDLSW;  WAIT FOR MEMORY OPERAND                7142.000
0DE1 0 0 0 2 3 C 0 0 00 0 000 3 3 2 0 0 0 3 2                RB=SHIFT(RD(SRA,-@04)),RAE=RMEXP; XFR DIVISOR TO RB    7143.000
0DE2 D 0 0 0 1 0 0 0 00 0 205 2 0 2 0 0 1 2 2                COMPLAN,RB=RA-RB,CORRZEROW,RAE=EXPK+RAE;COMPLEMENT B IF <0 7144.000
0DE3 8 0 0 0 1 0 0 0 00 0 004 0 0 0 0 1 1 1 2                RAE=RBE-RAE, OVFMOPEZ;                                 7145.000
0DE4 0 0 0 2 3 7 0 0 00 0 000 3 1 1 0 0 0 0 0                RA=SHIFT(RA(SRA,-@09));  LOAD DIVIDEND                 7146.000
0DE5 F 7 2 6 5 0 0 0 00 0 000 3 0 2 0 0 0 0 0                RB=SHIFT(RB(SLA,BITCNT)), *JUMPDV;  JUMP TO JDV02 PLUS 1 OR 3 7147.000
                                                    JDV02                                                           7148.000
0DE6 0 0 0 6 4 C 0 0 0B 0 000 3 1 1 4 0 0 0 0                RA=SHIFT(RA(SLA,BITCNT)), MPY(@C), INCRN;              7149.000
                                                    JDV02.01(JDV02+1)                                               7150.000
0DE7 0 5 0 0 0 0 0 4 00 0 0FB 0 0 0 3 0 0 0 0  DFB           RD=R(R) CPAOB, *GO TO JDV02.01.EXIT; REFETCH REGISTER OPERAND 7151.000
                                                    JDV02.02(JDV02+3)                                               7152.000
0DE9 0 0 0 2 4 2 0 0 00 0 000 3 0 2 0 0 0 0 0                RB=SHIFT(RB(SLA,@02)); D/4                             7153.000
0DEA 0 0 0 0 0 2 0 0 00 0 000 0 0 0 4 0 0 0 0                MPY(@2); <RP=3D/4> , <RPP=2D/4>                        7154.000
0DEB 0 0 0 2 3 C 0 0 00 0 000 3 0 2 0 0 0 0 3                RB=SHIFT(RB(SRA,-@04)), RXE=RAE+RBE;                   7155.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                                    FP  DIVIDE  DOUBLE
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                SPSHIFT(RD(SLL,@02)),RA=RA-RP,DIVLOOP,DECRN,IF %NCTRZ *HOP $,  7156.000
0DEC A 4 3 3 4 2 0 0 00 0 90C 2 2 0 6 0 0 2 3  DEC             RXE=EXPK+RBE;                                                  7157.000
0DED D 0 0 2 4 6 0 0 00 0 600 3 3 2 0 0 0 0 0                   RB=SHIFT(RD(SLA,@06)), MASK;                                   7158.000
0DEE F 7 1 0 1 0 0 0 00 0 007 2 0 1 0 0 0 0 0                   COMPLAF,RA=RA-RB, *JUMPNV;                                     7159.000
                                                        *** VECTORED JUMP FOR NORMALIZE (JUMP TO JNV12 PLUS 1, 3, OR 5)    7160.000
                                                        JNV12                                                              7161.000
0DEF D 0 0 2 4 0 0 0 00 0 100 3 1 0 0 0 0 0 0                   NOR2, NOD=SHIFT(RA(SLA,00));                                   7162.000
                                                        JNV12.01(JNV12+1)                                                  7163.000
                                                        R.RND.DIVIDE.DOUBLE.1                                              7164.000
0DF0 0 4 0 0 1 0 0 0 00 0 008 0 0 0 0 0 0 0 0  DF8              *GO TO JNV12.01EXIT;                                           7165.000
0DF1 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   *NOP;                                                          7166.000
                                                        JNV12.03(JNV12+3)                                                  7167.000
                                                        R.ZERO.DIVIDE.DOUBLE.1                                             7168.000
0DF2 0 0 5 4 2 3 7 2 0C 0 6E6 0 0 0 0 0 0 0 0                   MARIX=R(X)+I0,RSTRHF,DECODE(@1E),RESET(FPDWORD);               7169.000
0DF3 0 6 0 0 1 0 0 0 09 0 601 0 0 0 0 0 0 0 0  601              *GOTO FIRMWARE.FP,RSTFP;                                       7170.000
                                                        JNV12.05(JNV12+5)                                                  7171.000
                                                        R.NOR.DIVIDE.DOUBLE.1                                              7172.000
0DF4 0 0 0 4 5 0 0 0 00 0 000 3 1 1 0 0 0 0 0                   RA=SHIFT(RA(SLA,NORMCNT)); RESULT %NORMALIZED                  7173.000
0DF5 8 0 0 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0                   COKRMNG;                                                       7174.000
0DF6 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0                   R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;  7175.000
0DF7 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0                   R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                               7176.000
                                                                                                                           7177.000
                                                        JNV12.01EXIT                                                       7178.000
0DF8 D 0 0 0 1 0 0 0 00 0 700 0 0 0 0 0 1 2 2                   RAE=EXPK+RAE, PLUSONE;                                         7179.000
0DF9 D 0 1 2 3 9 9 0 19 0 580 3 1 4 0 0 1 2 0                   R(R)=SHIFT(RA(SRA,-@07)),ASFPW,NOD=EXPK+RAE,SETCC(#),FETCHPC;  7180.000
0DFA 0 3 0 3 5 9 B 6 00 0 000 3 1 4 0 0 0 0 0                   R(R0)=SHIFT(RA(SLL,@19)),*JUMPZ;                               7181.000
```

```
                SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                FP DIVIDE DOUBLE
PC   TSMAB+DR YX PCH+BDM+BAD ADDR

                                        JDV02.01.EXIT                                                    7182.000

0DFB 0 0 0 2 4 0 0 0 00 0 000 3 3 1 0 0 0 0 0          RA=SHIFT(RD(SLA,0));  TEST REG OPERAND SIGN        7183.000

0DFC D 0 1 2 4 0 9 0 19 0 580 3 1 4 0 0 0 0 0          R(R)=SHIFT(RA(SLA,0)), ASFPW, FFTCHPC, SETCC(#);   7184.000

0DFD 0 3 0 2 4 0 B 6 00 0 000 3 0 4 0 0 0 0 0          R(RO)=SHIFT(RB(SLA,0)), *JUMP7;                   7185.000
```

```
                SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                INSTRUCTION DECODE
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7186.000

                                                                                                                    7187.000
                                        ***                                                                         7188.000
                                        *** EXTENDED I/O INSTRUCTIONS - SECONDARY DECODE (Q=3F)                      7189.000
                                        ***                                                                         7190.000
                                        ***   THE REGISTER ADDRESSED BY THE INSTRUCTION BITS 16-31 IS                7191.000
                                        ***   ADDED TO IO BITS 16-31 TO FORM THE DEVICE CHANNEL AND                  7192.000
                                        ***   SUB-ADDRESS.  IF REGISTER 0 IS SPECIFIED BY THE INSTRUCTION            7193.000
                                        ***   THE CHANNEL AND SUB-ADDRESS IS TAKEN DIRECTLY FROM IO BITS             7194.000
                                        ***   16-31.  SUM BITS (ALU BITS) 17-23 ARE USED FOR THE CHANNEL             7195.000
                                        ***   ADDRESS AND SUM BITS 25-31 ARE USED FOR THE SUB-ADDRESS.               7196.000
                                        ***                                                                         7197.000

                                                                                                                    7198.000

                                        (@E00)                                                                      7199.000

                                        EXT.IO.INST                                                                 7200.000

0E00 4 4 0 1 0 D 0 0 02 C 702 0 0 0 0 0 0 0 0  E02      NOD=@007000000&T, IF MODE75S *GO TO $+2; FETCH CHNL & SUB ADDR     7201.000

0E01 0 6 0 0 1 0 0 0 00 0 7F7 0 0 0 0 0 0 0 0  7F7      *GO TO UNDEF.55; EXT.I/O UNDEFINED FOR 55 MODE                     7202.000

0E02 9 4 0 0 1 0 0 0 00 0 D84 0 0 0 0 0 0 0 0  E04      IF %EXFLAG *HOP $+2; EXECUTE MEM OR REG IS INVALID                 7203.000

0E03 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5      *GO TO UNDEF.75;                                                   7204.000

0E04 C 4 0 4 2 3 F 4 00 D 006 0 0 0 0 0 0 0 0  E06      T=R(R)+IO,OTHERBANK, IF NALUZ *GO TO $+2; CHANNEL & SUB ADDR IND   7205.000

0E05 0 0 0 0 2 0 F 0 00 0 000 0 0 0 0 0 0 0 0           T=IO; GET CHANNEL & SUB ADDR NOT INDEXED                           7206.000

0E06 0 0 0 7 0 B 1 0 00 2 000 0 0 0 0 0 0 0 0           S=%@FFFFFF00&T, TNIBL; GET SUB ADDR IN S REG 24-31                 7207.000

0E07 0 0 0 0 1 0 0 0 00 2 000 0 0 0 0 0 0 0 0           TNIBL; MOVE CHANNEL ADDR TO BITS 08-15                             7208.000

0E08 0 0 0 1 0 D F 0 02 0 7F0 0 0 0 0 0 0 0 0           T=@007F0000&T; GET CHANNEL ADDR IN T BITS 09-15                    7209.000

0E09 0 6 7 0 0 A F 0 00 0 2DB 0 0 0 0 0 0 0 0  2DB      T=S:T *LINK IO.REG.LOAD;                                           7210.000
                                        **                                                                          7212.000
                                        **   NOTE : IN THE ABOVE INSTRUCTION,T=T:S, THE CHANNEL                     7213.000
                                        **          AND SUB-CHANNEL ADDRESS ARE STORED IN THE                       7214.000
                                        **          T REGISTER BITS 08-15 AND BITS 24-31 RESPECTIVELY               7215.000
                                        **                                                                          7216.000
```

```
            S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                  I N S T R U C T I O N  D E C O D E
PC  T S M A B + D R  Y X PCH + R D M + B A D  ADDR
                                                                                                                    7217.000
                                        **                                                                          7219.000
                                        ** EXTENDED I/O REGISTERS                                                  7220.000
                                        **   ENTRY PARAMETERS :                                                    7221.000
                                        **     R(EDEV)= CHANNEL (DVC) ADDRESS IN BITS 9-15 & SUB-ADDR IN 25-31     7222.000
                                        **     R(RDEV)= SCRATCH PAD DEVICE ENTRY                                   7223.000
                                        **     R(INTR)= CHANNEL OR DEVICE INTERRUPT ENTRY ADDRESS                  7224.000
                                        **     R(INTRLOC)= SCRATCH PAD INTERRUPT ENTRY ADDRESS                     7225.000
                                        **     R(INTRTAB)= SCRATCHPAD INTERRUPT ENTRY                              7226.000
                                        **     R(SI)= SI MEMORY ADDRESS                                            7227.000
                                        **     R(IOCD.ADDR)= IOCD MEMORY ADDRESS OR STATUS LOCATIONS               7228.000
                                        **                   FOR EXTENDED I/O                                     7229.000
                                        **                                                                        7230.000


                                                                                                                    7231.000
0E0A 6 0 0 0 4 8 0 1 02 0 0F0 0 0 0 0 0 0 0 0                   NOD=@0F000000!FR(RDEV); TEST FOR CLASS F DEVICE             7232.000

0E0B 0 0 0 1 2 D 1 0 02 0 780 0 0 0 0 0 0 0 0                   S=@00780000&I0; GET AUGMENT OP-CODE BITS IN S REG BITS 9-12  7233.000

0E0C 2 4 0 0 1 0 0 0 01 0 78E 0 0 0 0 0 0 0 0  E0E              SHIFTS(SRL), IF ALU4-7Z *HOP $+2; BRANCH IF CLASS F DEVICE AND 7234.000
                                        *                         SHIFT AUGMENT CODE TO BITS 10-13.                 7236.000

0E0D 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5              *GO TO UNDEF.75; CLASS F NOT SPECIFIED, THUS UNDEFINED CLASS 7238.000

0E0E 0 0 0 0 4 0 6 0 01 0 080 0 0 0 0 0 0 0 0                   SHIFTS(SRL), DI=R(IOCD.ADDR); FETCH IOCD ADDR AND SHIFT      7239.000
                                        *                         AUGMENT CODE TO BITS 11-14.                       7241.000

0E0F 6 0 0 3 4 D F 0 02 0 FF0 0 0 0 0 0 0 0 0                   T=@000000FF&FR(EDEV); MASK SUB-CHANNEL ADDRESS BITS 23-29    7243.000

0E10 0 0 0 4 0 A 8 0 00 0 900 0 0 0 0 0 0 0 0                   R(RDEV)=R(RDEV):T; MERGE SUB-ADDRESS WITH DEVICE ENTRY       7244.000

0E11 0 0 0 0 1 E 8 0 00 0 B00 0 0 0 0 0 0 0 0                   R(TEMP1)=S; STORE AUGMENT CODE IN BITS 11/14                 7245.000

0E12 0 0 0 3 1 E 1 0 02 D E20 0 0 0 0 0 0 0 0                   S=(JMP.TBL↑-4&@00FF), OTHERBANK;GENERATE BASE ADDR OF JUMP TABLE 7246.000
                                        *                         IN BITS 24-31 OF S.                               7248.000

0E13 0 0 0 2 5 3 1 0 00 D B00 0 0 0 0 0 0 0 0                   S=SNIBL+R(TEMP1,HWS), OTHERBANK; SHIFT JUMP TABLE BASE ADDR TO 7250.000
                                        *  BITS 20-27 AND ADD AUGMENT CODE TO BITS 23-30.  (COMPUTE                7252.000
                                        *  JUMP TABLE ADDRESS).                                                    7253.000

0E14 6 0 0 4 4 D B 3 02 0 BF0 0 0 0 0 0 0 0 0                   FR(INTRTAB)=@BFFFFFFF&FR(INTRTAB); CLEAR WCS ENABLED FLAG    7255.000

0E15 6 0 1 4 3 D 6 0 02 0 000 0 0 0 0 0 0 0 0                   DI=@00FFFFFF&DI, *JUMPS; JUMP TO 'JMP.TBL' INDEXED BY AUGMENT 7256.000
                                        *  CODE (RIGHT SHIFTED 1).                                                 7258.000
```

```
                SYSTEM8 MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU                                                EXTENDED I/O BRI
PC   TSMAB+DR YX PCH+BDM+BAD ADDR

                                                                                                                    7259.000

                                                                                                                    7260.000
                                        ***                                                                         7261.000
                                        *** DISPATCH EXTENDED I/O DEACTIVATE                                        7262.000
                                        ***                                                                         7263.000
                                        ***   THE FOLLOWING SUB-ROUTINE IS USED BY THE BRI INSTRUCTION               7264.000
                                        ***   WHEN CLASS F CHANNELS ARE INDICATED BY THE POLLING                     7265.000
                                        ***   INTERRUPT LEVEL.                                                      7266.000
                                        ***                                                                         7267.000

                                                                                                                    7268.000

                                        DISPATCH.EXT.DEACTIVATE                                                     7269.000

0E16 0 0 0 0 4 0 6 0 00 0 000 0 0 0 0 0 0 0 0                   DI=R(IOCD.ADDR); FETCH CURRENT IOCD ADDRESS                7270.000

0E17 6 0 0 2 4 A B 4 02 0 300 0 0 0 0 0 0 0 0                   FR(OFFSET)=@00003000:FR(OFFSET); SET BRI RETURN & DACI FLAGS  7271.000

0E18 0 5 7 0 1 0 0 0 00 0 041 0 0 0 0 0 0 0 0  E41              *LINK BRI.DACI; GO SCHEDULE A DEACTIVATE CHANNEL           7272.000

0E19 6 5 0 0 4 0 F 3 00 0 921 0 0 0 0 0 0 0 0  E21              T=FR(INTRTAB), IF NCTRZ *GO TO EXT.BRI.RETURN; TEST FOR ERRORS  7273.000

0E1A 0 0 0 2 1 E F 0 02 0 200 0 0 0 0 0 0 0 0                   T=BRI.ERR.FLG;                                            7274.000

0E1B 0 6 0 0 0 A 1 0 00 0 57C 0 0 0 0 0 0 0 0  57C              S=S:T, *GO TO SYSTEM.CHECK.TRAP;                           7275.000

                                        EXT.ICT.PATCH                                                               7276.000

0E1C 1 4 0 0 1 0 0 0 00 0 7BE 0 0 0 0 0 0 0 0  E1E              IF IONORESP *HOP $+2;                                      7277.000

0E1D 0 5 0 0 1 0 0 0 00 0 0C9 0 0 0 0 0 0 0 0  EC9              *GO TO ICT.DRT+1;                                          7278.000

0E1E 0 0 0 0 1 E C 0 02 6 B00 0 0 0 0 0 0 0 0                   NU=(SCT.FLG:FINAL.FLG:N.RESP.FLG),SDEST;                   7279.000

0E1F 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                   NOD=N, *JUMPJ;                                             7280.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU JUMP TABLE IO INSTR

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7281.000
                                                                                                                    7283.000
                                                **                                                                  7284.000
                                                **      JUMP TABLE FOR EX.IO.INSTR FROM A JUMPS                     7285.000
                                                                                                                    7286.000
                                                ($+15&-16)                                                          7287.000
                                                JMP.TBL
0E20 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5      *GO TO UNDEF.75;                                           7288.000
                                                EXT.BRI.RETURN                                                      7289.000
0E21 0 6 0 0 1 0 0 0 00 0 437 0 0 0 0 0 0 0 0  437      *GO TO BRI.JUMP;                                           7290.000
                                                (JMP.TBL+2)                                                         7291.000
0E22 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5      *GO TO UNDEF.75;                                           7292.000
                                                SIO(JMP.TBL+4)                                                      7293.000
0E24 0 0 0 0 3 A 6 0 02 0 020 0 0 0 0 0 0 0 0           DI=@02000000:DI; FLAG BITS ORED WITH PARAMETERS            7294.000
0E25 6 5 0 0 4 0 0 3 00 0 048 0 0 0 0 0 0 0 0  E48      NOD=FR(INTRTAB), *GO TO DISPATCH.ARSTX.RSTX;                7295.000
                                                TIO(JMP.TBL+6)                                                      7296.000
0E26 0 0 0 0 3 A 6 0 02 0 040 0 0 0 0 0 0 0 0           DI=@04000000:DI; FLAG BITS ORED WITH PARAMETERS            7297.000
0E27 6 5 0 0 4 0 0 3 00 0 048 0 0 0 0 0 0 0 0  E48      NOD=FR(INTRTAB), *GO TO DISPATCH.ARSTX.RSTX;                7298.000
                                                STPIO(JMP.TBL+8)                                                    7299.000
0E28 0 0 0 0 3 A 6 0 02 0 050 0 0 0 0 0 0 0 0           DI=@05000000:DI; FLAG BITS ORED WITH PARAMETERS            7300.000
0E29 6 5 0 0 4 0 0 3 00 0 048 0 0 0 0 0 0 0 0  E48      NOD=FR(INTRTAB), *GO TO DISPATCH.ARSTX.RSTX;                7301.000
                                                RSCHNL(JMP.TBL+@A)                                                  7302.000
0E2A 0 0 0 0 3 A 6 0 02 0 200 0 0 0 0 0 0 0 0           DI=@20000000:DI; SET RESET CHANNEL                         7303.000
0E2B 0 5 0 0 1 0 0 0 00 0 0E5 0 0 0 0 0 0 0 0  EE5      *GO TO DISPATCH.WDOT;                                      7304.000
                                                HIO(JMP.TBL+@C)                                                     7305.000
0E2C 0 0 0 0 3 A 6 0 02 0 060 0 0 0 0 0 0 0 0           DI=@06000000:DI; FLAG BITS ORED WITH PARAMETERS            7306.000
0E2D 6 5 0 0 4 0 0 3 00 0 048 0 0 0 0 0 0 0 0  E48      NOD=FR(INTRTAB), *GO TO DISPATCH.ARSTX.RSTX;                7307.000
                                                GRIO(JMP.TBL+@E)                                                    7308.000
0E2E 0 0 0 0 3 A 6 0 02 0 090 0 0 0 0 0 0 0 0           DI=@09000000:DI; FLAG BITS ORED WITH PARAMETERS            7309.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    JUMP TABLE IO INSTR
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0E2F 6 5 0 0 4 0 0 3 00 0 048 0 0 0 0 0 0 0 0  E48        NOD=FR(INTRTAB), *GO TO DISPATCH.ARSTX.RSTX;                    7310.000
                                                  RSCTL(JMP.TBL+@10)                                                  7311.000
0E30 0 0 0 0 0 A 6 0 02 0 070 0 0 0 0 0 0 0 0             DI=@07000000:DI; FLAG BITS ORED WITH PARAMETERS              7312.000
                                                  ARSTX.RSTX.JUMP                                                     7313.000
0E31 0 5 0 0 1 0 0 0 00 0 048 0 0 0 0 0 0 0 0  E48        *GO TO DISPATCH.ARSTX.RSTX;                                  7314.000
                                                  ECWCS(JMP.TBL+@12)                                                  7315.000
0E32 6 0 0 0 4 A B 3 02 0 400 0 0 0 0 0 0 0 0             FR(INTRTAB)=@40000000:FR(INTRTAB); SET CHANNEL WCS ENABLED FLAG 7316.000
0E33 0 5 0 0 0 0 1 0 00 F 04C 0 0 0 0 0 0 0 0  E4C        S=T(ZE), *GO TO UPDATE.INT.TBL.EXIT+1; CLEAR CC'S AND EXIT   7317.000
                                                  (JMP.TBL+@14)                                                       7318.000
0E34 0 6 0 0 1 0 0 0 00 0 7F5 0 0 0 0 0 0 0 0  7F5        *GO TO UNDEF.75;                                             7319.000
                                                  WCWCS(JMP.TBL+@16)                                                  7320.000
0E36 6 0 0 0 4 D 0 4 02 0 400 0 0 0 0 0 0 0 0             NOD=@40000000&FR(OFFSET);TEST FOR CHANNEL WCS ENABLED FLAG    7321.000
0E37 0 5 0 0 1 0 0 0 00 0 060 0 0 0 0 0 0 0 0  E60        *GO TO WCWCS1;                                               7322.000
                                                  ECI(JMP.TBL+@18)                                                    7323.000
0E38 6 0 0 0 4 A B 3 02 0 010 0 0 0 0 0 0 0 0             FR(INTRTAB)=@01000000:FR(INTRTAB); SET ENABLED FLAG          7324.000
0E39 0 4 0 0 3 A 6 0 02 0 0C1 0 0 0 0 0 0 0 0  E31        DI=@0C000000:DI, *HOP ARSTX.RSTX.JUMP;                       7325.000
                                                  DCI(JMP.TBL+@1A)                                                    7326.000
0E3A 6 0 0 0 4 B B 3 02 0 030 0 0 0 0 0 0 0 0             FR(INTRTAB)=%@03000000&FR(INTRTAB); CLEAR ENABLED AND RI PENDING 7327.000
0E3B 0 4 0 0 3 A 6 0 02 0 0D1 0 0 0 0 0 0 0 0  E31        DI=@0D000000:DI, *HOP ARSTX.RSTX.JUMP;                       7328.000
                                                  ACI(JMP.TBL+@1C)                                                    7329.000
0E3C 6 0 0 0 4 A B 3 02 0 040 0 0 0 0 0 0 0 0             FR(INTRTAB)=@04000000:FR(INTRTAB); SET INTERRUPT ACTIVE FLAG 7330.000
0E3D 0 4 0 0 1 E 6 0 02 0 0E1 0 0 0 0 0 0 0 0  E31        DI=@0E000000, *HOP ARSTX.RSTX.JUMP;                          7331.000
                                                  DACI(JMP.TBL+@1E)                                                   7332.000
0E3E 6 0 0 2 4 A B 4 02 0 A00 0 0 0 0 0 0 0 0             FR(OFFSET)=@0000A000:FR(OFFSET); SET DACI FLAG               7333.000
0E3F 0 0 0 0 1 0 0 0 00 0 704 0 0 0 0 0 0 0 0             RESET(ENAINTFF); NEXT INSTRUCTION WILL BE NON-INTERRUPTABLE  7334.000
0E40 6 0 0 0 4 B B 3 02 0 040 0 0 0 0 0 0 0 0             FR(INTRTAB)=%@04000000&FR(INTRTAB); CLEAR ACTIVE FLAG        7335.000
                                                  BRI.DACI                                                            7336.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    J U M P  T A B L E  I O  I N S T R
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0E41 0 0 0 3 1 E 6 0 02 0 080 0 0 0 0 0 0 0 0          DI=#000000008; DEACTIVATE INTERRUPT                                7337.000

                                                    DISPATCH.AICT.ICT                                                     7338.000

0E42 0 5 7 0 4 0 0 0 00 0 BA9 0 0 0 0 0 0 0 0  EA9     NOD=R(INTRTAB) *LINK AICT.ICT;                                     7339.000

0E43 0 0 0 0 4 0 F 0 00 0 C00 0 0 0 0 0 0 0 0          T=R(OFFSET); TEST FOR BRI RETURN FLAG (BIT 19)                     7340.000

0E44 5 1 0 0 0 0 0 0 00 8 000 0 0 0 0 0 0 0 0          NOD=T, IF %BMUX19 *JUMPJ; EXIT IF BRI FLAG                         7341.000

0E45 0 0 0 0 1 0 0 0 00 0 501 0 0 0 0 0 0 0 0          SETCC(S);                                                          7342.000

                                                    N.COUNTER.ERROR.CODE                                                  7343.000

0E46 A 4 0 0 0 0 0 0 00 0 D0F 0 0 0 0 0 0 0 0  E4F     NOD=T,IF %NCTR0 *GOTO DACI.EXIT; TEST FOR FATAL ERROR FLG (NCTR0) 7344.000

0E47 0 6 0 0 1 0 0 0 00 0 57C 0 0 0 0 0 0 0 0  57C     *GO TO SYSTEM.CHECK.TRAP;                                          7345.000

                                                    DISPATCH.ARSTX.RSTX                                                   7346.000

0E48 0 5 7 0 4 0 0 0 00 0 B79 0 0 0 0 0 0 0 0  E79     NOD=R(INTRTAB) *LINK ARSTX.RSTX;                                   7347.000

0E49 E 4 0 0 4 0 F 4 00 0 906 0 0 0 0 0 0 0 0  E46     T=FR(OFFSET), IF %NCTRZ *GO TO N.COUNTER.ERROR.CODE;               7348.000

0E4A 0 0 0 0 4 0 1 0 00 0 300 0 0 0 0 0 0 0 0          S=R(STATUS.CC); RESTORE STATUS & CC'S IN S (RIGHT SHIFTED 1)       7349.000

                                                    UPDATE.INT.TBL.EXIT                                                   7350.000

0E4B 0 0 0 0 4 0 F 0 00 0 B00 0 0 0 0 0 0 0 0          T=R(INTRTAB);                                                      7351.000

0E4C 0 0 0 4 0 0 4 0 00 A D00 0 0 0 0 0 0 0 0          SCRATCH(R(INTRLOC))=T; SAVE FLAG BITS RAM LOADED                   7352.000

                                                    SETCC.EXIT                                                            7353.000

0E4D 0 0 0 0 1 E 0 0 19 0 581 0 0 0 0 0 0 0 0          NOD=S, SETCC(S),FETCHPC;  SET CONDITION CODE & FETCH ($+8)         7354.000

0E4E 0 3 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0          RESET(HIREG), *JUMPZ;                                              7355.000

                                                    DACI.EXIT                                                             7356.000

0E4F 3 5 0 0 1 0 0 0 00 A 04D 0 0 0 0 0 0 0 0  E4D     IF BMUX18 *GO TO SETCC.EXIT;                                       7357.000

0E50 0 6 0 0 1 0 0 0 00 0 F80 0 0 0 0 0 0 0 0  F80     *GO TO NON.INTERRUPTABLE.EXIT;                                     7358.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU EXTEND I/O SEL BUS TRANSFERS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                    7359.000
                                        **                                                          7361.000
                                        ** EXTENDED I/O SEL BUS TRANSFERS                          7362.000
                                        **     ENTRY PARAMETERS:                                    7363.000
                                        **       FULLMAR = DESTINATION BUS DATA AS FOLLOWS:         7364.000
                                        **                                                          7365.000
                                        **     MAR BITS *0          7*8              15*16      23* 7366.000
                                        **     BUS BITS *0          7*8              15*16      23* 7367.000
                                        **              ********************************************* 7368.000
                                        ** DEST BUS     * %INTERRUPT * CHANNEL PHYSICAL * CHANNEL SUB* 7369.000
                                        **              *  LEVEL     * ADDRESS          * ADDRESS    * 7370.000
                                        **              ********************************************* 7371.000
                                        **                                                          7372.000
                                        **     T=ARSTX, RSTX, AICT, ICT, OR WDOT FUNCTION CODE      7373.000
                                        **                                                          7374.000

                                                                                                    7375.000
                                        (@E51)                                                      7376.000
                                        **                                                          7377.000
                                        **     ISSUE ARSTX                                          7378.000
                                        **                                                          7379.000
                                        EXT.CMD.ARSTX                                               7380.000
0E51 0 4 0 0 1 0 0 0 1F 0 043 0 0 0 0 0 0 0 0 E53       NOD=MAR, ARSTX, *HOP ADVANCE.ON.BUS;        7381.000
                                                                                                    7382.000
                                        **                                                          7383.000
                                        **     ISSUE AICT                                           7384.000
                                        **                                                          7385.000
                                        EXT.CMD.AICT                                                7386.000
0E52 0 0 0 0 1 0 0 0 1F 0 010 0 0 0 0 0 0 0 0           NOD=MAR, AICT;                              7387.000
                                                                                                    7388.000
                                        ADVANCE.ON.BUS                                              7389.000
0E53 0 0 0 0 0 0 0 0 00 0 000 0 0 0 0 0 0 0 0           NOD=T; ADVANCE TRANSFER SHOULD BE ON THE BUS 7390.000
0E54 0 4 0 0 1 0 0 0 00 0 00A 0 0 0 0 0 0 0 0 E5A       *GO TO EXT.TRANSFER.EXIT;  CHANNEL RESPONSE SHOLUD BE ON THE BUS 7391.000
                                        **                                                          7392.000
                                        **     ISSUE RSTX                                           7393.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    EXTEND I/O SEL BUS TRANSFERS
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                **                                                  7394.000
                                                EXT.CMD.RSTX                                        7395.000
0E55 0 4 0 0 1 0 0 0 1C 0 048 0 0 0 0 0 0 0 0  E58      NOD=MAR, RSTX, *HOP FINAL.ON.BUS;          7396.000
                                                                                                    7397.000
                                                **                                                  7398.000
                                                **      ISSUE ICT                                   7399.000
                                                **                                                  7400.000
                                                EXT.CMD.ICT                                         7401.000
0E56 0 4 0 0 1 0 0 0 1C 0 018 0 0 0 0 0 0 0 0  E58      NOD=MAR, ICT, *HOP FINAL.ON.BUS;           7402.000
                                                                                                    7403.000
                                                **                                                  7404.000
                                                **      ISSUE WDOT                                  7405.000
                                                **                                                  7406.000
                                                EXT.CMD.WDOT                                        7407.000
0E57 0 0 0 0 1 0 0 0 1E 0 000 0 0 0 0 0 0 0 0           NOD=MAR, WDOT;                              7408.000
                                                FINAL.ON.BUS                                        7409.000
0E58 0 0 0 0 0 0 0 0 00 0 000 0 0 0 0 0 0 0 0           NOD=T; FINAL TRANSFER SHOULD BE ON THE BUS  7410.000
                                                RESPONSE.ON.BUS                                     7411.000
0E59 0 0 0 0 0 0 0 0 00 0 000 0 0 0 0 0 0 0 0           NOD=T;  CHANNEL RESPONSE SHOULD BE ON THE BUS 7412.000
                                                EXT.TRANSFER.EXIT                                   7413.000
0E5A 0 1 0 0 5 0 0 0 00 0 700 0 0 0 0 0 0 0 0           NOD=R(BUSREQ,HWS), *JUMPJ; FETCH BUS TRANSFER CODE 7414.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    EXT I/O RAM LOAD
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7415.000

                                                                                                                    7416.000
                                          **                                                                        7417.000
                                          **     EXTENDED I/O RAM LOAD SEQUENCE                                     7418.000
                                          **                                                                        7419.000
                                          **                                                                        7420.000

                                                                                                                    7421.000

                                          EXT.RAM.LOAD                                                              7422.000

0E5B 6 0 0 2 4 D 1 1 02 0 FFD 0 0 0 0 0 0 0 0          S=@0000FF00:FR(RDEV), CLDNU; GET PHYSICAL ADDR               7423.000

0E5C 0 0 0 3 1 E 1 0 00 0 000 0 0 0 0 0 0 0 0          S=SNIBR; SHIFT PHYSICAL ADDR & TEST FOR ZERO                 7424.000

0E5D 6 0 0 0 4 A B 3 02 0 800 0 0 0 0 0 0 0 0          FR(INTRTAB)=@80000000:FR(INTRTAB); SET RAM LOADED FLAG       7425.000

0E5E B 5 0 3 1 E F 0 00 0 089 0 0 0 0 0 0 0 0  E89     T=SNIBR, IF ALU7 *GO TO EXT.PHY.ADDR.0; ALIGN PHYSICAL ADDR  7426.000

                                          *           IN BITS 24-31 AND EXIT IF PHYSICAL ADDRESS =0                7427.000

0E5F 0 1 0 0 0 A F 0 02 0 080 0 0 0 0 0 0 0 0          T=@08000000:T, *JUMPJ;  GENERATE ARSTX                       7428.000

                                          WCWCS1                                                                    7429.000

0E60 C 4 0 0 3 A 6 0 02 0 0B3 0 0 0 0 0 0 0 0  E63     DI=@0B000000:DI, IF NALUZ *HOP WCWCS2; SET WCWCS BUS CODE AND 7430.000
                                          *      BRANCH IF WCS ENABLED FLAG =1.                                     7432.000

0E61 0 0 0 1 1 E 1 0 02 0 010 0 0 0 0 0 0 0 0          S=CHAN.WCS.N.FLG; SET CPU ERROR CODE                         7434.000

0E62 0 6 0 0 1 0 0 0 00 0 57C 0 0 0 0 0 0 0 0  57C     *GO TO SYSTEM.CHECK.TRAP;                                    7435.000

                                          WCWCS2                                                                    7436.000

0E63 0 5 0 0 1 0 0 0 00 0 048 0 0 0 0 0 0 0 0  E48     *GO TO DISPATCH.ARSTX.RSTX;                                  7437.000

                                          EXT.RSTX.RAM.LOAD                                                         7438.000

0E64 6 0 0 0 4 A B 4 02 0 800 0 0 0 0 0 0 0 0          FR(OFFSET)=@80000000:FR(OFFSET); SET RAM LOAD FLAG           7439.000

0E65 9 4 0 0 0 0 C 0 00 F DB7 0 0 0 0 0 0 0 0  E67     NU=T(ZE), IF %IOTIMEOUT *HOP $+2;                            7440.000

0E66 0 0 0 0 1 E C 0 02 6 A20 0 0 0 0 0 0 0 0          NU=(SCT.FLG:FINAL.FLG:IO.TIM.FLG), SDEST;                    7441.000

0E67 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=N, *JUMPJ; RETURN                                        7442.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU DEVICE CLASS F INTERRUPT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7443.000

                                                                                                                    7444.000
                                                    **                                                              7445.000
                                                    ** DEVICE CLASS F INTERRUPT                                     7446.000
                                                    **                                                              7447.000

                                                                                                                    7448.000

                                                    (@E68)                                                          7449.000

                                                    DEV.CLASSF.INT                                                  7450.000

0E68 0 0 0 0 4 0 F 0 00 0 000 0 0 0 0 0 0 0 0                 T=R(N.PSW2);                                      7451.000

0E69 0 0 0 0 0 0 0 0 00 0 106 0 0 0 0 0 0 0 0                 NOD=T, RESET(UNBLOCK);                            7452.000

0E6A 5 5 7 0 0 0 0 0 00 8 073 0 0 0 0 0 0 0 0  E73            NOD=T, IF %BMUX16 *LINK CURRENT.PSW;              7453.000

0E6B 3 4 0 0 0 0 0 0 00 9 00D 0 0 0 0 0 0 0 0  E6D            NOD=T, IF BMUX17 *GO TO FORMAT.ACK.ACTIVATE;       7454.000

                                                    FORMAT.ACK.DEACTIVATE                                           7455.000

0E6C 0 4 0 3 1 E 6 0 02 0 01F 0 0 0 0 0 0 0 0  E6F            DI=@00000001 *HOP $+3; ACKNOWLEDGE AND DEACTIVATE INTERRUPT  7456.000

                                                    FORMAT.ACK.ACTIVATE                                             7457.000

0E6D 0 0 0 0 1 0 0 0 00 0 906 0 0 0 0 0 0 0 0                 SET(UNBLOCK); ALLOW EXTERNAL INTERRUPTS           7458.000

0E6E 0 0 0 3 1 E 6 0 02 0 020 0 0 0 0 0 0 0 0                 DI=@00000002; ACKNOWLEDGE AND ACTIVATE INTERRUPT  7459.000

0E6F 0 5 7 0 4 0 0 0 00 0 BA9 0 0 0 0 0 0 0 0  EA9            NOD=R(INTRTAB) *LINK AICT.ICT;                    7460.000

0E70 E 4 0 0 4 0 F 3 00 0 906 0 0 0 0 0 0 0 0  E76            T=FR(INTRTAB), IF %NCTRZ *GO TO CLASSF.INT.ERROR;  7461.000

0E71 0 0 0 4 0 0 4 0 00 A D00 0 0 0 0 0 0 0 0                 SCRATCH(R(INTRLOC))=T; UPDATE INT TABLE ENTRY IF BLOCKED MODE  7462.000

                                                    INT.EXIT                                                        7463.000

0E72 0 6 0 0 1 0 0 0 00 8 06A 0 0 0 0 0 0 0 0   6A            PCTOMAR, *GO TO SCHEDULE.75.TRAP;                 7464.000

                                                    CURRENT.PSW                                                     7465.000

0E73 0 0 0 1 1 0 1 0 02 A 900 0 0 0 0 0 0 0 0                 S=SCRATCH(@90); FETCH CURRENT PSW                 7466.000

0E74 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0                 T=S;                                              7467.000

0E75 0 1 0 0 0 0 0 0 00 0 000 0 0 0 0 0 0 0 0                 NOD=T, *JUMPJ;                                    7468.000

                                                    CLASSF.INT.ERROR                                                7469.000

0E76 0 0 0 0 1 E F 0 02 0 C00 0 0 0 0 0 0 0 0                 T=(EXT.IO.FLG:INT.ERR.FLG); SET INTERRUPT SEQUENCE ERROR FLAG  7470.000

0E77 0 6 0 0 0 A 1 0 00 0 57D 0 0 0 0 0 0 0 0  57D            S=S:T, *GO TO SYSTEM.CHECK.STATUS; EXIT TO SYSTEM CHECK TRAP  7471.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    ARSTX  RSTX
PC  TSMAB+DR YX PCH+BDM+BAD ADDR

                                                                                    7472.000
                                        ***       ARSTX   RSXT       ***           7473.000
                                        ***                                        7474.000
0E78 0 0 0 0 4 0 0 0 00 0 B00 0 0 0 0 0 0 0 0       NOD=R(INTRTAB);                 7475.000
                                        ARSTX.RSTX                                 7476.000
0E79 5 4 0 0 3 0 8 0 00 4 70F 0 0 0 0 0 0 0 0 E7F   R(BUSREQ)=DI, IF %BMUX00 *GO TO EXT.RAM.LOADED; TEST RAM LOAD  7477.000
                                        *                                          7478.000
0E7A 0 5 7 0 1 0 0 0 00 0 058 0 0 0 0 0 0 0 0 E5B   *LINK EXT.RAM.LOAD; RAM IS NOT LOADED DO RAM LOAD SEQUENCE  7479.000
0E7B 0 4 7 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0 E7F   *LINK EXT.RAM.LOADED; RAM HAS BEEN LOADED BY SCHEDULER DO ARSTX  7480.000
0E7C E 1 0 0 4 0 F 3 00 0 900 0 0 0 0 0 0 0 0       T=FR(INTRTAB), IF %NCTR7 *JUMPJ; ERROR DETECTION N-COUNTER  7481.000
                                        *                                          7482.000
0E7D 0 0 0 4 0 0 4 0 00 A D00 0 0 0 0 0 0 0 0       SCRATCH(R(INTRLOC))=T; STORE FLAG BIT RAM LOAD  7483.000
0E7E 0 0 0 0 4 0 F 0 00 0 700 0 0 0 0 0 0 0 0       T=R(BUSREQ); RESTORE ORIGINAL REQUEST & GO DO ARSTX RSTX  7484.000
                                        EXT.RAM.LOADED                             7485.000
0E7F 1 4 0 0 1 0 0 0 00 0 7A1 0 0 0 0 0 0 0 0 E81   IF IORESPRDY *HOP $+2; BRANCH IF READY  7486.000
0E80 0 6 7 0 0 1 C 0 00 F 329 0 0 0 0 0 0 0 0 329   NU=%T(ZE), *LINK COUNT.DOWN; GO CLEAR PENDING READY  7487.000
0E81 0 0 0 0 1 E C 0 02 0 120 0 0 0 0 0 0 0 0       NU=@12000000; LOAD N-COUNTER FOR 144 CLOCKS (21.6 MICRO.SEC.)  7488.000
                                        ARSTX                                      7489.000
                                        ARSTX.RETRY                                7490.000
0E82 0 5 7 0 4 0 E 0 00 0 951 0 0 0 0 0 0 0 0 E51   FULLMAR=R(RDEV), *LINK EXT.CMD.ARSTX;  7491.000
0E83 9 4 0 0 1 0 0 0 00 0 AB8 0 0 0 0 0 0 0 0 E88   IF %IORETRY:IOCHBUSY *GO TO YY.Y;  7492.000
                                        TEST.COUNT.1                               7493.000
0E84 2 4 0 0 1 0 0 0 15 0 906 0 0 0 0 0 0 0 0 E86   CLRTO,IF NCTR7 *GO TO $+2;     7494.000
0E85 0 4 3 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0 E82   DECRN *GO TO ARSTX.RETRY;       7495.000
0E86 0 0 0 0 1 E C 0 02 6 080 0 0 0 0 0 0 0 0       NU=CC4, SDEST; SET CHANNEL BUSY (CC4)  7496.000
                                        EXT.DUD                                    7497.000
0E87 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0       *JUMPJ;                        7498.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    ARSTX RSTX
PC   TSMAB+DR YX PCH+BDM+BAD ADDR

                                                  YY.Y                                                                    7499.000
0E88 9 4 0 0 1 0 0 0 00 0 78A 0 0 0 0 0 0 0 0 E8A         IF %IONORESP *GO TO $+2;                                        7500.000
                                                  EXT.PHY.ADDR.0                                                          7501.000
0E89 0 4 0 0 1 E C 0 02 6 107 0 0 0 0 0 0 0 0 E87         NU=CC3, SDEST, *GO TO EXT.DUD; NO RESPONSE SET CC3              7502.000
0E8A 0 0 0 0 1 E C 0 02 0 560 0 0 0 0 0 0 0 0             NU=@56000000; LOAD N-COUNTER FOR 258 CLOCKS (38.7 MICRO.SEC.)   7503.000
                                                  RETRY.READY                                                             7504.000
0E8B 1 4 0 0 1 0 0 0 00 0 7AF 0 0 0 0 0 0 0 0 E8F         IF IORESPRDY *GO TO RSTX;                                       7505.000
                                                  TEST.COUNT.2                                                            7506.000
0E8C 2 4 0 0 1 0 0 0 00 0 90E 0 0 0 0 0 0 0 0 E8E         IF NCTRZ *GO TO $+2; ERROR                                      7507.000
0E8D 0 4 3 0 1 0 0 0 00 0 00B 0 0 0 0 0 0 0 0 E8B         DECRN *GO TO RETRY.READY;                                       7508.000
0E8E 0 4 0 0 1 E C 0 02 6 847 0 0 0 0 0 0 0 0 E87         NU=(SCT.FLG:RDY.TIM.FLG), SDEST, *GO TO EXT.DUD;                7509.000
                                                  **                                                                      7510.000
                                                  RSTX                                                                    7511.000
0E8F 0 0 0 0 1 E C 0 02 0 210 0 0 0 0 0 0 0 0             NU=@21000000; LOAD N-COUNTER FOR 264 CLOCKS (39.6 MICRO.SEC.)   7512.000
                                                  RSTX.RETRY                                                              7513.000
0E90 0 5 7 0 1 0 0 0 00 0 055 0 0 0 0 0 0 0 0 E55         NOD=MAR, *LINK EXT.CMD.RSTX;                                    7514.000
0E91 9 4 0 0 1 0 0 0 00 0 EB6 0 0 0 0 0 0 0 0 E96         IF %IORETRY *GO TO RSTX.DRT;                                    7515.000
                                                  TEST.COUNT.3                                                            7516.000
0E92 2 4 0 0 1 0 0 0 00 0 904 0 0 0 0 0 0 0 0 E94         IF NCTRZ *GO TO $+2;                                            7517.000
0E93 0 4 3 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0 E90         DECRN *GO TO RSTX.RETRY;                                        7518.000
0E94 0 0 0 0 1 E C 0 02 6 A10 0 0 0 0 0 0 0 0             NU=(SCT.FLG:FINAL.FLG:CNT.EXH.FLG),SDEST; SYSTEM CHECK TRAP     7519.000
                                                  EXT.DUD.1                                                               7520.000
0E95 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *JUMPJ;                                                         7521.000
                                                  **                                                                      7522.000
                                                  RSTX.DRT                                                                7523.000
0E96 9 4 0 0 1 0 0 0 00 0 BB8 0 0 0 0 0 0 0 0 E98         IF %IOCHBUSY *GO TO $+2;                                        7524.000
0E97 0 4 0 0 1 E C 0 02 6 A85 0 0 0 0 0 0 0 0 E95         NU=(SCT.FLG:FINAL.FLG:BUSY.FLG),SDEST, *HOP EXT.DUD.1;          7525.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                          A R S T X   R S T X
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0E98 0 0 0 0 4 0 0 0 12 0 C00 0 0 0 0 0 0 0 0          NOD=R(OFFSET),SAVFSIGN;  TEST RAM LOAD                                  7526.000
0E99 0 0 0 0 4 0 F 0 00 0 600 0 0 0 0 0 0 0 0          T=R(SI.VECTOR); FETCH ADDRESS                                           7527.000
0E9A 0 0 0 3 0 3 E 0 02 0 140 0 0 0 0 0 0 0 0          FULLMAR=@00000014+T; BIAS ADDRESS                                       7528.000
0E9B 5 5 0 0 3 0 F 0 00 1 064 0 0 0 0 0 0 0 0 E64      T=DI IF %SIGNSAVE *GO TO EXT.RSTX.RAM.LOAD;                             7529.000
0E9C 0 0 0 0 3 0 1 0 1E 1 080 0 0 0 0 0 0 0 0          S=DI, WRITE,FRCWORD; STORE STATUS                                       7530.000
0E9D 0 4 0 0 1 0 0 0 01 0 08F 0 0 0 0 0 0 0 0 E9F      SHIFTS(SRL), *HOP $+2; POSITION CC                                      7531.000
0E9E 0 0 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *NOP; REQUIRED FOR ALIGNMENT (NOT EXECUTED)                             7532.000
0E9F 9 4 0 0 0 0 C 0 00 F DB1 0 0 0 0 0 0 0 0 EA1      NU=T(ZE), IF %IOTIMEOUT *GO TO RSTX.TRANSFER.OK; CLEAR N REG            7533.000
0EA0 0 4 0 0 1 E C 0 02 6 A27 0 0 0 0 0 0 0 0 EA7      NU=(SCT.FLG:FINAL.FLG:IO.TIM.FLG), SDEST, *HOP EXT.DUD.4;               7534.000
                                                  RSTX.TRANSFER.OK                                                          7535.000
0EA1 0 0 0 0 4 D F 0 02 0 7F0 0 0 0 0 0 0 0 0          T=@7F000000&R(BUSREQ); FETCH BUS TRANSFER CODE                          7536.000
0EA2 0 0 0 0 0 8 0 0 02 0 0E0 0 0 0 0 0 0 0 0          NOD=@0E000000!T; TEST FOR ACI                                           7537.000
0EA3 0 0 0 0 1 E 8 0 00 0 300 0 0 0 0 0 0 0 0          R(STATUS.CC)=S;  SAVE STATUS WORD RIGHT SHIFTED FOR SETCC(S)            7538.000
0EA4 C 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          IF NALUZ *JUMPJ; EXIT IF NOT ACI                                        7539.000
0EA5 0 0 0 1 1 0 1 0 02 A 850 0 0 0 0 0 0 0 0          S=SCRATCH(@85); FETCH CURRENT ACTIVE INTERRUPT COUNT                    7540.000
0EA6 0 5 0 0 1 0 0 0 00 0 0E0 0 0 0 0 0 0 0 0 EE0      *GO TO INCREMENT.ACTIVE.COUNT; GO BUMP ACTIVE COUNT                     7541.000
                                                  EXT.DUD.4                                                                 7542.000
0EA7 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *JUMPJ; DRT TIMEOUT ERROR EXIT                                          7543.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU AICT-ICT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7544.000
                                                                                                                    7545.000
                                                                    ***            AICT   ICT
                                                                    (@EA8)                                          7546.000
0EA8 0 0 0 0 4 0 0 0 00 0 B00 0 0 0 0 0 0 0 0                           NOD=R(INTRTAB);                             7547.000
                                                                    AICT.ICT                                        7548.000
0EA9 5 4 0 0 3 0 8 0 00 4 70F 0 0 0 0 0 0 0 0  EAF                      R(BUSREQ)=DI, IF %BMUX00 *GO TO EXT.RAM.LOADED1; TEST RAM LOAD  7549.000
                                                                    *                                               7550.000
0EAA 0 5 7 0 1 0 0 0 00 0 05B 0 0 0 0 0 0 0 0  E5B                      *LINK EXT.RAM.LOAD; RAM IS NOT LOADED DO RAM LOAD SEQUENCE  7551.000
0EAB 0 5 7 0 1 0 0 0 00 0 07F 0 0 0 0 0 0 0 0  E7F                      *LINK EXT.RAM.LOADED; SCHEDULE RAM LOAD ARSTX/RSTX  7552.000
0EAC E 1 0 0 4 0 F 3 00 0 900 0 0 0 0 0 0 0 0                           T=FR(INTRTAB), IF %NCTRZ *JUMPJ; ERROR DETECTION N-COUNTER  7553.000
                                                                    *                                               7554.000
0EAD 0 0 0 4 0 0 4 0 00 A D00 0 0 0 0 0 0 0 0                           SCRATCH(R(INTRLOC))=T; SAVE FLAG BIT RAM LOAD  7555.000
0EAE 0 0 0 0 4 0 F 0 00 0 700 0 0 0 0 0 0 0 0                           T=R(BUSREQ); RESTORE ORIGINAL REQUEST & GO DO THE AICT ICT  7556.000
                                                                    EXT.RAM.LOADED1                                 7557.000
0EAF 1 4 0 0 1 0 0 0 00 0 7A1 0 0 0 0 0 0 0 0  EB1                      IF IORESPRDY *HOP $+2; BRANCH IF NOT READY PENDING  7558.000
0EB0 0 6 7 0 0 1 C 0 00 F 329 0 0 0 0 0 0 0 0  329                      NU=%T(ZE), *LINK COUNT.DOWN; GO CLEAR PENDING READY  7559.000
0EB1 0 0 0 0 1 E C 0 02 0 450 0 0 0 0 0 0 0 0                           NU=@45000000; LOAD N-COUNTER FOR 552 CLOCKS (82.8 MICR.SECS.)  7560.000
                                                                    AICT                                            7561.000
                                                                    AICT.RETRY                                      7562.000
0EB2 0 5 7 0 4 0 E 0 00 0 952 0 0 0 0 0 0 0 0  E52                      FULLMAR=R(RDEV), *LINK EXT.CMD.AICT;        7563.000
0EB3 1 4 0 0 1 0 0 0 00 0 A85 0 0 0 0 0 0 0 0  EB5                      IF IORETRY:IOCHBUSY *GO TO TEST.COUNT1.1;   7564.000
0EB4 0 4 0 0 1 0 0 0 00 0 009 0 0 0 0 0 0 0 0  EB9                      *GO TO NO.RETRY.NOT.BUSY;                   7565.000
                                                                    TEST.COUNT1.1                                   7566.000
0EB5 2 4 0 0 1 0 0 0 00 0 907 0 0 0 0 0 0 0 0  EB7                      IF NCTRZ *GO TO $+2;                        7567.000
0EB6 0 4 3 0 1 0 0 0 00 0 002 0 0 0 0 0 0 0 0  EB2                      DECRN *GO TO AICT.RETRY;                    7568.000
0EB7 0 0 0 0 1 E C 0 02 6 890 0 0 0 0 0 0 0 0                           NU=(SCT.FLG:BUSY.FLG:CNT.EXH.FLG), $DEST;   7569.000
                                                                    EXT.DUD.2                                       7570.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                                          A I C T - I C T
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0EB8 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0                *JUMPJ;                                                    7571.000

                                                          NO.RETRY.NOT.BUSY                                          7572.000

0EB9 9 4 0 0 1 0 0 0 00 0 7BB 0 0 0 0 0 0 0 0  EBB        IF %IONORESP *GO TO $+2;                                   7573.000

0EBA 0 4 0 0 1 E C 0 02 6 108 0 0 0 0 0 0 0 0  EB8        NU=CC3, SDEST, *GO TO EXT.DUD.2; SET (CC3)                 7574.000

0EBB 0 0 0 0 1 E C 0 02 0 580 0 0 0 0 0 0 0 0              NU=@58000000; LOAD N-COUNTER FOR 264 CLOCKS (39.6 MICR.SECS.) 7575.000

                                                          RETRY.READY1                                               7576.000

0EBC 9 4 0 0 1 0 0 0 00 0 7AE 0 0 0 0 0 0 0 0  EBE        IF %IORESPRDY *GO TO $+2;                                  7577.000

0EBD 0 5 0 0 1 0 0 0 00 0 0C2 0 0 0 0 0 0 0 0  EC2        *GO TO ICT;                                                7578.000

                                                          TEST.COUNT2                                                7579.000

0EBE 2 5 0 0 1 0 0 0 00 0 9C0 0 0 0 0 0 0 0 0  EC0        IF NCTRZ *GO TO $+2;                                       7580.000

0EBF 0 5 3 0 1 0 0 0 00 0 0BC 0 0 0 0 0 0 0 0  EBC        DECRN *GO TO RETRY.READY1;                                 7581.000

0EC0 0 0 0 0 1 E C 0 02 6 840 0 0 0 0 0 0 0 0              NU=(SCT.FLG:RDY.TIM.FLG), SDEST; SYSTEM CHECK TRAP         7582.000

                                                          EXT.DUD.3                                                  7583.000

0EC1 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPJ;                                                    7584.000

                                                          **                                                         7585.000

                                                          ICT                                                        7586.000

0EC2 0 0 0 0 1 E C 0 02 0 210 0 0 0 0 0 0 0 0              NU=@21000000; LOAD N-COUNTER FOR 258 CLOCKS (38.7 MICR.SECS.) 7587.000

                                                          ICT.RETRY                                                  7588.000

0EC3 0 5 7 0 1 0 0 0 00 0 056 0 0 0 0 0 0 0 0  E56        NOD=MAR, *LINK EXT.CMD.ICT;                                7589.000

0EC4 9 4 0 0 1 0 0 0 00 0 AB8 0 0 0 0 0 0 0 0  EC8        IF %IORETRY:IOCHBUSY *GO TO ICT.DRT;                       7590.000

                                                          TEST.COUNT3                                                7591.000

0EC5 2 4 0 0 1 0 0 0 00 0 907 0 0 0 0 0 0 0 0  EC7        IF NCTRZ *GO TO $+2;                                       7592.000

0EC6 0 4 3 0 1 0 0 0 00 0 003 0 0 0 0 0 0 0 0  EC3        DECRN *GO TO ICT.RETRY;                                    7593.000

0EC7 0 4 0 0 1 E C 0 02 6 A91 0 0 0 0 0 0 0 0  EC1        NU=(SCT.FLG:FINAL.FLG:CNT.EXH.FLG:BUSY.FLG),SDEST,*HOP EXT.DUD.3; 594.000

                                                          ICT.DRT                                                    7595.000

0EC8 0 5 0 0 4 0 F 0 00 0 61C 0 0 0 0 0 0 0 0  E1C        T=R(SI.VECTOR), *GO TO EXT.ICT.PATCH; FETCH ADDRESS         7596.000

0EC9 0 0 0 3 0 3 E 0 02 0 140 0 0 0 0 0 0 0 0              FULLMAR=@00000014+T; BIAS ADDRESS                          7597.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                                    A I C T - I C T
PC   T S M A B + D R  Y X PCH + R D M + B A D  ADDR

0ECA 0 0 0 0 4 0 0 0 00 0 C00 0 0 0 0 0 0 0 0          NOD=R(OFFSET); TEST FOR BRI RETURN & DACI FLAGS (BITS 19 & 18)   7598.000

0EC8 5 4 0 0 3 0 1 0 00 A 00D 0 0 0 0 0 0 0 0  ECD     S=DI, IF %BMUX18 *GO TO $+2; WAIT FOR DRT & BRANCH IF DACI       7599.000

0ECC 0 0 0 0 3 0 F 0 1F 1 080 0 0 0 0 0 0 0 0          T=DI, WRITE, FRCWORD; STORE STATUS                               7600.000

0ECD 0 0 0 0 4 0 6 0 00 0 700 0 0 0 0 0 0 0 0          DI=R(BUSREQ); FETCH INITIAL BUS REQUEST                          7601.000

0ECE 0 0 0 3 3 8 0 0 02 0 010 0 0 0 0 0 0 0 0          NOD=@00000001!DI; TEST FOR ACK-DEACTIVATE ICT                    7602.000

                                                EXT.ICT.CK.TIMEOUT                                                      7603.000

0ECF 9 4 0 0 1 0 0 0 00 0 DB2 0 0 0 0 0 0 0 0  ED2     IF %IOTIMEOUT *HOP ICT.TRANSFER.OK;                              7604.000

0ED0 0 0 0 0 1 E C 0 02 6 A20 0 0 0 0 0 0 0 0          NU=(SCT.FLG:FINAL.FLG:IO.TIM.FLG), SDEST;                        7605.000

                                                DUD.EDX                                                                 7606.000

0ED1 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0          *JUMPJ;                                                          7607.000

                                                ICT.TRANSFER.OK                                                         7608.000

0ED2 8 4 0 0 4 0 0 0 00 0 C0C 0 0 0 0 0 0 0 0  EDC     NOD=R(OFFSET), IF ALUZ *GO TO EXT.UPDATE.INTR.FLAGS; EXIT IF     7609.000
*                                                               ACK-DEACTIVATE                                          7611.000

0ED3 3 4 0 0 1 0 0 0 00 A 00E 0 0 0 0 0 0 0 0  EDE     IF BMUX18 *GO TO AICT.ICT.EXIT;BRANCH IF NOT DACI (IF ACK-ACTIV) 7613.000

                                                EXT.DACI.EXIT                                                           7614.000

0ED4 0 0 0 1 1 0 1 0 02 A 850 0 0 0 0 0 0 0 0          S=SCRATCH(@85); FETCH CURRENT ACTIVE COUNT                       7615.000

0ED5 0 0 0 0 1 4 F 0 00 0 000 0 0 0 0 0 0 0 0          T=S-1;                                                           7616.000

0ED6 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0          NOD=@04000000&FR(OFFSET); TEST FOR PREVIOUSLY ACTIVE FLAG        7617.000

0ED7 2 4 0 0 0 0 0 0 00 0 80B 0 0 0 0 0 0 0 0  EDB     NOD=T, IF ALUNEG *GO TO RESET.INTERRUPT.ACTIVE; EXIT IF CNT <0   7618.000

0ED8 8 4 0 0 0 0 0 0 00 0 00C 0 0 0 0 0 0 0 0  EDC     NOD=T, IF ALUZ *GO TO EXT.UPDATE.INTR.FLAGS;EXIT IF NOT ACTIVE   7619.000

0ED9 8 0 0 0 0 0 0 0 00 0 504 0 0 0 0 0 0 0 0          NOD=T, IF ALU7 RESET(DINTRA);IF ACT CNT =0, RESET ACTIVE         7620.000

                                                EXT.UPDATE.ACTIVE.COUNT                                                 7621.000

0EDA 0 4 0 1 0 0 4 0 02 A 85C 0 0 0 0 0 0 0 0  EDC     SCRATCH(@85)=T, *HOP EXT.UPDATE.INTR.FLAGS;                      7622.000

                                                RESET.INTERRUPT.ACTIVE                                                  7623.000

0ED8 0 0 0 0 1 0 0 0 00 0 504 0 0 0 0 0 0 0 0          RESET(DINTRA); TURN OFF INTERRUPT ACTIVE LITE                    7624.000

                                                EXT.UPDATE.INTR.FLAGS                                                   7625.000

0EDC 0 0 0 0 4 0 F 0 00 0 800 0 0 0 0 0 0 0 0          T=R(INTRTAB);                                                    7626.000

0EDD 0 0 0 4 0 0 4 0 00 A D00 0 0 0 0 0 0 0 0          SCRATCH(R(INTRLOC))=T; UPDATE INTERRUPT ENTRY                     7627.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU AICT-ICT

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                AICT.ICT.EXIT                                                          7628.000

0EDE 0 0 0 0 0 0 C 0 00 F 000 0 0 0 0 0 0 0 0          NU=T(ZE);                                                       7629.000

0EDF 0 1 0 3 1 E 1 0 02 0 000 0 0 0 0 0 0 0 0          S=0, *JUMPJ;                                                    7630.000

                                                                                                                       7631.000
                                                *                                                                      7632.000
                                                *    INCREMENT ACTIVE COUNT IS USED ONLY BY ACI INSTRUCTIONS.           7633.000
                                                *                                                                      7634.000

                                                                                                                       7635.000

                                                INCREMENT.ACTIVE.COUNT                                                 7636.000

0EE0 6 0 0 0 4 D 0 4 02 0 040 0 0 0 0 0 0 0 0          NOD=@04000000&FR(OFFSET); TEST FOR PREVIOUSLY ACTIVE            7637.000

0EF1 0 0 0 0 1 6 F 0 00 0 000 0 0 0 0 0 0 0 0          T=S+1; INCREMENT ACTIVE COUNT                                   7638.000

0EE2 C 5 0 0 0 0 0 0 00 0 0DE 0 0 0 0 0 0 0 0  EDE     NOD=T, IF NALUZ *GO TO AICT.ICT.EXIT; EXIT IF ALREADY ACTIVE    7639.000

0EE3 0 0 0 0 0 0 C 0 00 F D04 0 0 0 0 0 0 0 0          NU=T(ZE), SET(DINTRA); SET INTERRUPT ACTIVE LITE & CLEAR N REG  7640.000

0EE4 0 1 0 1 0 0 4 0 02 A 850 0 0 0 0 0 0 0 0          SCRATCH(@85)=T, *JUMPJ; UPDATE ACTIVE COUNT & EXIT              7641.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU RESET CHANNEL
PC T SMAB+DR YY PCH+ROM+BAD ADDR

```
                                                                                                                    7642.000
                                                    **                                                              7644.000
                                                    **   RESET CHANNEL ROUTINE -WDOT-  **                           7645.000
                                                    **                                                              7646.000
                                                    (@EE5)                                                          7647.000

                                                    DISPATCH.WDOT                                                   7649.000

0EE5 0 0 0 0 4 0 E 0 00 0 900 0 0 0 0 0 0 0 0           FULLMAR=R(RDEV); SET CHANNEL AND SUB-ADDRESSES IN MAR       7650.000

0EE6 6 0 0 4 4 D B 3 02 0 000 0 0 0 0 0 0 0 0           FR(INTRTAB)=@00FFFFFF&FR(INTRTAB); CLEAR INTERRUPT TABLE FLAGS  7651.000

0EE7 0 0 0 0 3 0 F 0 1E 0 000 0 0 0 0 0 0 0 0           T=DI, WDOT; EXECUTE FIRST WDOT PSCHNL                       7652.000

0EE8 0 0 0 0 4 0 E 0 00 0 902 0 0 0 0 0 0 0 0           FULLMAR=R(RDEV), FORCEZ;  SET HARWARE WAIT FOR READY FLAG   7653.000

0EE9 0 5 7 0 1 0 0 0 00 0 059 0 0 0 0 0 0 0 0 E59       *LINK RESPONSE.ON.BUS;  WAIT FOR CHANNEL RESPONSE SIGNAL    7654.000

0EEA 9 4 0 0 1 0 0 0 00 0 7BC 0 0 0 0 0 0 0 0 EEC       IF %IONORESP *GO TO $+2;                                    7655.000

0EEB 6 0 0 2 4 A B 4 02 0 800 0 0 0 0 0 0 0 0           FR(OFFSET)=@00008000:FR(OFFSET); SET FLAG IONORESP          7656.000

0EEC 0 0 0 0 1 E C 0 02 0 800 0 0 0 0 0 0 0 0           NU=@80000000; LOAD N-COUNTER WITH COUNT DOWN READY (38.4MICR.S)  7657.000

0EED 0 4 7 0 4 0 F 0 00 0 C0F 0 0 0 0 0 0 0 0 EEF       T=R(OFFSET), *LINK PSCHNL.WAIT.READY;                       7658.000

0EFE 0 5 0 0 1 0 0 0 00 0 04B 0 0 0 0 0 0 0 0 E4B       *GO TO UPDATE.INT.TRL.EXIT;                                 7659.000

                                                    RSCHNL.WAIT.READY                                               7660.000

0EFF 1 4 3 0 0 0 0 0 00 0 7A7 0 0 0 0 0 0 0 0 EF7       NOD=T, DECRN, IF IORESPRDY *GO TO READY.EXIT;               7661.000

0EF0 A 5 0 0 1 0 0 0 00 0 9EF 0 0 0 0 0 0 0 0 EEF       IF %NCTRZ *GO TO RSCHNL.WAIT.READY;                         7662.000

0EF1 6 0 0 2 0 A B 4 02 0 400 0 0 0 0 0 0 0 0           FR(OFFSET)=@00004000:T; SET NOT READY FLAG                  7663.000

0EF2 5 4 0 0 1 0 0 0 00 8 00B 0 0 0 0 0 0 0 0 EF8       IF %BMUX16 *GO TO CHANNEL.NOT.PRESENT;                      7664.000

                                                    *RESPONSE.BUT.NOT.READY                                         7665.000

0EF3 0 4 0 0 1 E F 0 02 0 845 0 0 0 0 0 0 0 0 EF5       T=(EXT.IO.FLG:RDY.TIM.FLG), *HOP $+2;                       7666.000

                                                    CATASTROPHIC.ERROR                                              7667.000

0EF4 0 0 0 0 1 E F 0 02 0 A00 0 0 0 0 0 0 0 0           T=(EXT.IO.FLG:FINAL.FLG);                                   7668.000

0EF5 0 0 0 1 0 A C 0 02 6 020 0 0 0 0 0 0 0 0           NU=RSTCHNL.FLG : T, SDEST; SET SYSTEM CHECK TRAP ERROR FLAGS  7669.000

0EF6 0 6 0 0 1 0 0 0 00 0 57C 0 0 0 0 0 0 0 0 57C       *GO TO SYSTEM.CHECK.TRAP;                                   7670.000

                                                    READY.EXIT                                                      7671.000

0EF7 0 5 7 0 3 0 F 0 00 0 057 0 0 0 0 0 0 0 0 E57       T=DI, *LINK EXT.CMD.WDOT; GO EXECUTE WDOT FINAL TRANSFER    7672.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
                                                                    R E S E T   C H A N N E L
S E L   3 2 / 7 5   C P U
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0EF8 1 4 0 0 1 0 0 0 00 0 7B4 0 0 0 0 0 0 0 0  EF4        IF IONORESP *GO TO CATASTROPHIC.ERROR;                                  7673.000

0EF9 6 0 0 3 1 E B 4 02 6 000 0 0 0 0 0 0 0 0              FR(OFFSET)=@00000000, SDEST; CLEAR ALL ERROR FLAGS AND SAVE IN S 7674.000

0EFA 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0              *JUMPJ; GOOD WDOT TRANSFER EXIT                                         7675.000

                                                    CHANNEL.NOT.PRESENT                                                        7676.000

0EFB 0 1 0 0 1 E C 0 02 6 100 0 0 0 0 0 0 0 0              NU=CC3, SDEST, *JUMPJ; SET NO RESPONSE CC3 (CHANNEL NOT PRESENT)  7677.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    CLASS F IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                7678.000
                                        ($+1)                                   7679.000
                                        CHECK.CLASSF.IPL                        7681.000
0EFD 0 0 0 2 3 D F 0 02 0 F00 0 0 0 0 0 0 0 0       T=@0000F000&DI; GET DVC CLASS FROM DRT                  7682.000
0EFE 0 0 0 2 0 8 0 0 02 0 F00 0 0 0 0 0 0 0 0       NOD=@0000F000!T; CHECK FOR CLASS F                      7683.000
0EFF 0 0 0 1 3 0 4 0 02 A 8F0 0 0 0 0 0 0 0 0       SCRATCH(@8F)=DI; SAVE 'IDENTIFY PROTOCOL' DRT           7684.000
0F00 C 1 0 0 3 0 0 0 00 0 705 0 0 0 0 0 0 0 0       NOD=DI, RESET(FLAG), IF NALUZ *JUMPJ; RETURN IF NOT CLASS F 7685.000
0F01 0 0 0 0 1 E E 0 02 0 705 0 0 0 0 0 0 0 0       FULLMAR=@70000000, RESET(FLAG); CLEAR MAR               7686.000
0F02 0 0 0 0 1 E F 0 02 0 020 0 0 0 0 0 0 0 0       T=@02000000; SET IOCD1 FOR READ TO 000000               7687.000
0F03 0 6 7 0 1 0 0 0 00 0 300 0 0 0 0 0 0 0 0 300   *LINK MEMORY.WRITE; GO STORE AT LOCATION 0             7688.000
0F04 0 0 0 0 1 E F 0 02 0 600 0 0 0 0 0 0 0 0       T=@60000000; SET CC & SIL FLAGS FOR IOCD2               7689.000
0F05 6 0 0 3 0 A B 0 02 0 A00 0 0 0 0 0 0 0 0       FR(ICL.IOCD2)=@000000A0:T; SET IOCD2 XFER CNT =@A0 (BYT CNT 120) 7690.000
0F06 0 0 0 3 1 3 E 0 02 0 040 0 0 0 0 0 0 0 0       FULLMAR=@4+MAR; INCREMENT TO LOCATION 4                 7691.000
0F07 0 6 7 0 1 0 0 0 00 0 300 0 0 0 0 0 0 0 0 300   *LINK MEMORY.WRITE; GO STORE AT LOCATION 4             7692.000
0F08 6 0 0 5 4 D B 1 02 0 000 0 0 0 0 0 0 0 0       FR(RDEV)=@FF00FFFF&FR(RDEV); SET IPL PSEUDO INT LVL = @7F 7693.000
0F09 0 0 0 0 1 E 6 0 02 0 030 0 0 0 0 0 0 0 0       DI=@03000000; SET IPL ARSTX/RSTX BUS CODE (IOCD ADDR = 0) 7694.000
0F0A 0 6 7 0 1 0 0 0 00 0 E78 0 0 0 0 0 0 0 0 E78   *LINK ARSTX.RSTX-1; GO SCHEDULE RAM LOAD & IPL ARSTX/RSTX 7695.000
0F0B A 5 0 0 1 E F 0 00 0 92E 0 0 0 0 0 0 0 0 F2E   T=S, IF %NCTRZ *GO TO EXT.IPL.ERROR;                    7696.000
0F0C 0 0 0 0 1 0 0 0 00 0 F04 0 0 0 0 0 0 0 0       SET(ENAINTFF);ENABLE EXTERNAL INTERRUPTS                7697.000
                                        TEST.IOCD2                              7698.000
0F0D 0 0 0 3 1 E E 0 02 0 040 0 0 0 0 0 0 0 0       FULLMAR=@00000004; SET MEMORY LOCATION 4 (WORD 1)       7699.000
0F0E 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0 30B   *LINK MEMORY.READ; FETCH IOCD2 FROM LOCATION 4          7700.000
0F0F 0 0 0 4 3 8 0 0 00 0 800 0 0 0 0 0 0 0 0       NOD=R(ICL.IOCD2)!DI; CURRENT IOCD2 = INITIAL IOCD2 ?    7701.000
0F10 9 4 0 0 1 0 0 0 00 0 DA2 0 0 0 0 0 0 0 0 F12   IF %FFINT *GO TO $+2; CONTINUE IF NO INTERRUPT         7702.000
0F11 0 5 0 0 1 0 0 0 00 0 030 0 0 0 0 0 0 0 0 F30   *GO TO IPL.EARLY.INT.ERR;                              7703.000
0F12 8 5 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0 F0D   IF ALUZ *GO TO TEST.IOCD2;WAIT FOR IOCD2 IN MEM TO BE OVERLAYED 7704.000
```

```
          S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    C L A S S  F  I P L
PC  T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7705.000

                                                                                                                    7706.000
                                        ***                                                                         7707.000
                                        *** NOTE:  AT THIS TIME THE EXT I/O IPL DATA TRANSFER SHOULD                 7708.000
                                        ***        BE IN PROGRESS.                                                  7709.000
                                        ***                                                                         7710.000

                                                                                                                    7711.000

                                        TEST.CLASSF.IPL.INT                                                         7712.000

0F13 0 0 0 3 1 E E 0 02 0 000 0 0 0 0 0 0 0 0                 FULLMAR=@00000000; SET MEMORY LOCATION 0              7713.000

0F14 0 6 7 0 1 0 0 0 00 0 308 0 0 0 0 0 0 0 0  308            *LINK MEMORY.READ; GO FETCH NEW PSW 1 FROM LOCATION 0 7714.000

0F15 0 0 0 0 3 0 8 0 00 0 200 0 0 0 0 0 0 0 0                 R(PSW1)=DI; SAVE NEW PSW 1                            7715.000

0F16 9 4 0 0 6 0 0 0 00 0 DA6 0 0 0 0 0 0 0 0  F16            NOD=INTLVL, IF *FFINT *HOP $; WAIT FOR EXT I/O INTERRUPT 7716.000

                                                                                                                    7717.000
                                        ***                                                                         7718.000
                                        ***  NOTE:  AT THIS TIME THE EXT I/O IPL DATA TRANSFER SHOULD               7719.000
                                        ***         BE COMPLETE.  MEMORY LOCATION 0 SHOULD CONTAIN THE NEW          7720.000
                                        ***         PSW WORD 1 AND LOCATION 4 SHOULD CONTAIN NEW PSW WORD           7721.000
                                        ***         2.  THESE LOCATIONS WERE LOADED BY THE IPL DATA TRANSFER.       7722.000
                                        ***                                                                         7723.000

                                                                                                                    7724.000

0F17 0 0 0 1 6 D F 0 02 0 7F0 0 0 0 0 0 0 0 0                 T=@007F0000&INTLVL; GET POLLING LEVEL                 7725.000

0F18 0 0 0 1 0 8 0 0 02 0 7F0 0 0 0 0 0 0 0 0                 NOD=@007F0000!T; POLLING LEVEL =7F ? (IPL PSEUDO LVL) 7726.000

0F19 0 0 0 0 1 0 0 0 00 0 106 0 0 0 0 0 0 0 0                 RESET(UNBLOCK);  BLOCK EXTERNAL LEVELS                7727.000

0F1A C 5 0 0 1 0 0 0 00 0 02D 0 0 0 0 0 0 0 0  F2D            IF NALUZ *GO TO EXT.IPL.ERROR.1;                      7728.000

0F1B 0 0 0 3 1 E 6 0 02 0 010 0 0 0 0 0 0 0 0                 DI=@00000001; SET AICT/ICT ACK. INT & DEACTIVATE BUS CODE 7729.000

0F1C 0 6 7 0 1 0 0 0 00 0 EA8 0 0 0 0 0 0 0 0  EA8            *LINK AICT.ICT-1; GO SCHEDULE AICT/ICT                7730.000

0F1D A 5 0 0 1 E F 0 00 0 92E 0 0 0 0 0 0 0 0  F2E            I=S, IF %NCTRZ *GO TO EXT.IPL.ERROR;                  7731.000

                                                                                                                    7732.000
                                        ***                                                                         7733.000
                                        ***  NOTE:  AT THIS TIME THE EXT I/O CHANNEL SHOULD HAVE POSTED             7734.000
                                        ***         TERMINATION STATUS IN MEMORY LOCATIONS 0 AND 4.                 7735.000
                                        ***                                                                         7736.000

                                                                                                                    7737.000

0F1E 0 0 0 0 1 E 6 0 02 0 0D0 0 0 0 0 0 0 0 0                 DI=@0D000000; SET DISABLE CHANNEL INT ARSTX/RSTX BUS CODE 7738.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  32/75  C P U                                                                    C L A S S   F   I P L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0F1F 0 6 7 0 1 0 0 0 00 0 E78 0 0 0 0 0 0 0 0  E78        *LINK ARSTX.RSTX-1; GO SCHEDULE ARSTX/RSTX DCI                    7739.000
0F20 A 4 0 0 1 E F 0 00 0 90E 0 0 0 0 0 0 0 0  F2E        T=S, IF %NCTRZ *GO TO EXT.IPL.ERROR;                             7740.000
                                                       CHECK.CHANNEL.IPL.STATS                                          7741.000
0F21 0 0 0 3 1 E E 0 02 0 040 0 0 0 0 0 0 0 0             FULLMAR=@00000004; SET MEMORY ADDR 4                          7742.000
0F22 0 6 7 0 1 0 0 0 00 0 30B 0 0 0 0 0 0 0 0  30B        *LINK MEMORY.READ; GO FETCH STATUS WORD 2                        7743.000
0F23 0 0 0 0 1 E F 0 02 0 1F0 0 0 0 0 0 0 0 0             T=@1F000000; BUILD CHANNEL STATUS MASK                        7744.000
0F24 0 0 0 1 0 A 1 0 02 0 D30 0 0 0 0 0 0 0 0             S=@00D30000:T; SET STATUS MASK = @1FD30000                    7745.000
0F25 0 0 0 0 3 D 0 0 00 0 000 0 0 0 0 0 0 0 0             NOD=S&DI; TEST CHANNEL STATUS WORD 2 FOR IPL ERRORS           7746.000
0F26 0 0 0 0 4 0 F 0 00 0 200 0 0 0 0 0 0 0 0             T=R(PSW1); FETCH NEW PSW 1                                    7747.000
0F27 C 4 0 0 1 0 0 0 00 0 00D 0 0 0 0 0 0 0 0  F2D        IF NALUZ *GO TO EXT.IPL.ERROR.1; EXIT IF BAD STATUS           7748.000
0F28 4 4 0 0 1 0 0 0 00 A 00D 0 0 0 0 0 0 0 0  F2D        IF ENBL.AEXP *GO TO EXT.IPL.ERROR.1; EXIT IF EARLY INT ERR FLAG  7749.000
0F29 6 0 0 4 0 D 2 0 00 D 004 0 0 0 0 0 0 0 0             PC=FR(PCMASK)&T, RESET(HIREG), OTHERBANK; LOAD PC FROM PSW 1   7750.000
0F2A 0 0 0 2 1 E 6 0 02 0 400 0 0 0 0 0 0 0 0             DI=@00004000; GENERATE THE ACTUAL PSW 2 (INTS ARE BLOCKED)     7751.000
0F2B 0 0 0 1 3 0 4 0 02 A 900 0 0 0 0 0 0 0 0             SCRATCH(@90)=DI;              SAVE PSD2                       7752.000
0F2C 0 6 0 0 1 0 0 0 00 0 9C6 0 0 0 0 0 0 0 0  9C6        *GO TO LPSD2; GO FILL PIPELINE AND BEGIN EXECUTE AT MACRO LVL    7753.000
                                                                                                                         7754.000
                                                       EXT.IPL.ERROR.1                                                  7755.000
0F2D 0 4 0 3 1 E F 0 02 0 00E 0 0 0 0 0 0 0 0  F2E        T=@00000000, *HOP EXT.IPL.ERROR; CLEAR STATUS WORD               7756.000
                                                       EXT.IPL.ERROR                                                    7757.000
0F2E 0 0 0 2 1 E 1 0 02 0 040 0 0 0 0 0 0 0 0             S=IPL.ERR.FLG;  SET CPU STATUS ERR FLG                         7758.000
0F2F 0 6 0 0 0 A 1 0 00 0 5B9 0 0 0 0 0 0 0 0  5B9        S=S:T, *GO TO M.55.TRAP.HALT;                                    7759.000
                                                       IPL.EARLY.INT.ERR                                                7760.000
0F30 0 0 0 0 1 0 0 0 00 0 B06 0 0 0 0 0 0 0 0             SET(ENBL.AEXP);  SET EARLY INTERRUPT ERROR FLAG                7761.000
0F31 0 5 0 0 1 0 0 0 00 0 013 0 0 0 0 0 0 0 0  F13        *GO TO TEST.CLASSF.IPL.INT; GO PROCESS INTERRUPT                 7762.000
                                                                                                                         7763.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    CLASS F IPL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                MAP.MODE.INVALID.ERR                                    7764.000

0F32 0 6 7 0 1 0 0 0 00 0 5C4 0 0 0 0 0 0 0 0  5C4      *LINK RESTORE.CPU.MODE;  GO RESTORE INITIAL CPU MODE    7765.000

0F33 0 6 7 0 1 0 0 0 00 8 59C 0 0 0 0 0 0 0 0  59C      PCTOMAR, *LINK DECREMENT.PC; GO BACK DATE PC            7766.000

0F34 0 6 0 0 1 0 0 0 00 0 557 0 0 0 0 0 0 0 0  557      *GO TO MAP.FAULT.TRAP;                                  7767.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S F L   3 2 / 7 5   C P U                                                    W C S   S U B - R O U T I N E
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7768.000
                                                    ***                                                             7770.000
                                                    *** THE FOLLOWING SUB-ROUTINES MUST NOT BE MOVED WITHOUT         7771.000
                                                    *** NOTIFING ALL WCS USERS                                       7772.000
                                                    ***                                                             7773.000

                                                                                                                    7774.000
                                                    ***                                                             7775.000
                                                    *** WRITE WCS                                                   7776.000
                                                    ***    ENTRY PARAMETERS:                                        7777.000
                                                    ***      S=WCS ADDRESS                                          7778.000
                                                    ***      T=WCS DATA                                             7779.000
                                                    ***                                                             7780.000

                                                                                                                    7781.000
                                                    (@F38)                                                          7782.000

                                                    WRITE.WCS                                                       7783.000

0F38 0 0 0 0 1 E 0 0 00 0 805 0 0 0 0 0 0 0 0             NOD=S,SET(ENAUORD);                                       7784.000

0F39 0 0 0 0 0 0 0 0 00 7 005 0 0 0 0 0 0 0 0             WCS(S)=T,RESET(ENAUORD);WRITE TO WCS                      7785.000

0F3A 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *JUMPJ;RETURN                                             7786.000

                                                                                                                    7787.000
                                                    ***                                                             7788.000
                                                    *** READ WCS                                                    7789.000
                                                    ***    ENTRY PARAMETER:                                         7790.000
                                                    ***      S=WCS ADDRESS                                          7791.000
                                                    ***                                                             7792.000
                                                    ***    EXIT PARAMETER:                                          7793.000
                                                    ***      S=WCS DATA                                             7794.000
                                                    ***                                                             7795.000

                                                                                                                    7796.000
                                                    (@F38)                                                          7797.000

                                                    READ.WCS                                                        7798.000

0F3B 0 0 0 0 1 E 0 0 00 0 805 0 0 0 0 0 0 0 0             NOD=S,SET(ENAUORD);                                       7799.000

0F3C 0 0 0 0 1 0 1 0 00 8 005 0 0 0 0 0 0 0 0             S=WCS(S),RESET(ENAUORD);READ FROM WCS                     7800.000

0F3D 0 1 0 0 1 0 0 0 00 0 000 0 0 0 0 0 0 0 0             *JUMPJ;RETURN                                             7801.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU POWER UP SEQUENCE

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7802.000
                                                    POWER.UP.ERROR                                                  7803.000
                                                    POWER.UP.S.P.CLEAR                                              7804.000
0F3E 0 0 0 1 0 0 4 0 02 A 8D0 0 0 0 0 0 0 0 0                 SCRATCH(@8D)=T;  SAVE POWER UP ERROR FLAGS          7805.000
0F3F 0 6 7 0 1 0 0 0 00 0 120 0 0 0 0 0 0 0 0  120           *LINK SYSTEM.RESET;                                 7806.000
0F40 0 6 7 0 1 0 0 0 00 0 26F 0 0 0 0 0 0 0 0  26F           *LINK PP.CSW.ADDR;                                  7807.000
0F41 0 6 7 0 1 0 0 0 00 0 267 0 0 0 0 0 0 0 0  267           *LINK PP.CSW.ADDR1;                                 7808.000
0F42 9 4 0 0 1 0 0 0 00 0 F47 0 0 0 0 0 0 0 0  F47           IF %SERIAL.PANEL *GO TO POWER.UP.EXIT;              7809.000
0F43 0 6 7 0 0 1 C 0 00 F AF3 0 0 0 0 0 0 0 0  AF3           NH=%T(ZE), *LINK S.P.CLEAR; SEND ONE SERIAL FRAME (ZEROS)  7810.000
0F44 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0                 T=0; CLEAR STOP FLAGS                               7811.000
0F45 0 0 0 0 1 0 0 0 00 0 804 0 0 0 0 0 0 0 0                 SFT(HIREG);                                         7812.000
0F46 0 6 7 0 1 0 0 0 00 0 C22 0 0 0 0 0 0 0 0  C22            *LINK S.P.PSW.RD+1; GOTO INITIAL POWER UP PSW TO PANEL  7813.000
                                                    POWER.UP.EXIT                                                   7814.000
0F47 0 0 0 1 1 0 1 0 02 A 8D0 0 0 0 0 0 0 0 0                 S=SCRATCH(@8D); FETCH POWER UP ERROR FLAGS          7815.000
0F48 0 0 0 0 1 E 0 0 00 0 000 0 0 0 0 0 0 0 0                 NOD = S; TEST POWER UP FLAGS                        7816.000
0F49 0 0 0 3 1 E F 0 02 0 000 0 0 0 0 0 0 0 0                 T=0; CLEAR ADDRESS STOP FLAGS                       7817.000
0F4A 2 4 0 1 0 0 4 0 02 A 8DC 0 0 0 0 0 0 0 0  F4C           SCRATCH(@8D)=T, IF ALUNEG *HOP $+2;                 7818.000
0F4B 0 6 0 0 1 0 0 0 00 0 110 0 0 0 0 0 0 0 0  110           *GO TO PWR.OFF.OUT; EXIT TO POWER UP TRAP           7819.000
0F4C 0 0 0 1 1 E F 0 02 0 800 0 0 0 0 0 0 0 0                 T=UP.PE.FLG; SET OPERAND PARITY ERROR FLAG          7820.000
0F4D 0 0 0 3 0 A F 0 02 6 100 0 0 0 0 0 0 0 0                 T=PWR.F.UP.FLG:T, SDEST;                            7821.000
0F4E 0 6 0 0 1 0 0 0 00 0 5B9 0 0 0 0 0 0 0 0  5B9           *GO TO M.55.TRAP.HALT;                              7822.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SFL 32/75 CPU SERIAL PANEL CHECK MAP ERRORS

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7823.000
                                                        *                                                           7824.000
                                                        ($+1)                                                       7825.000
                                                        S.P.CHECK.MAP.ERROR                                         7826.000
0F50 1 4 0 0 1 0 0 0 00 0 763 0 0 0 0 0 0 0 0  F53              IF PROTV:MAPINVALID *HOP S.P.MAP.ERROR;             7827.000
0F51 9 1 0 0 0 0 0 0 00 0 550 0 0 0 0 0 0 0 0                   NOD=T, IF %OPNORESP:OPTIMEOUT:OPRNOPE *JUMPJ;       7828.000
0F52 0 6 0 0 0 0 0 0 00 0 AD3 0 0 0 0 0 0 0 0  AD3              NOD=T, *GO TO S.P.MEMERR;                           7829.000
                                                        S.P.MAP.ERROR                                               7830.000
0F53 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0                   S=SCRATCH(@88); GET CURRENT C-DISPLAY               7831.000
0F54 0 0 0 0 1 E F 0 00 0 805 0 0 0 0 0 0 0 0                   T=S, SET(ENAUORD);                                  7832.000
0F55 0 0 0 3 0 0 0 0 02 0 010 0 0 0 0 0 0 0 0                   NOD=@00000001&T; TEST FOR MEMORY ADDRESS (MA) BIT   7833.000
0F56 1 4 0 0 1 0 0 0 00 0 F6C 0 0 0 0 0 0 0 0  F5C              IF MAPINVALID *GO TO S.P.M.INVALID;                 7834.000
                                                        S.P.M.PROTECT                                               7835.000
0F57 B 5 0 0 1 0 0 0 00 4 060 0 0 0 0 0 0 0 0  F60              RSTPROTV, IF ALUZ *GO TO S.P.SET.ERR.9;             7836.000
0F58 0 0 0 1 1 0 1 0 02 A 880 0 0 0 0 0 0 0 0                   S=SCRATCH(@88); GET CURRENT B DISPLAY               7837.000
0F59 0 0 0 0 1 0 E 0 00 0 005 0 0 0 0 0 0 0 0                   FULLMAR=MAR, RESET(ENAUORD); REMAP MAR              7838.000
0F5A 0 0 0 0 1 E F 0 1E 1 083 0 0 0 0 0 0 0 0                   T=S, WRITE, FRCWORD, IGNSTOP;                       7839.000
0F5B 0 6 0 0 1 0 0 0 00 0 ACF 0 0 0 0 0 0 0 0  ACF              *GO TO S.P.BIBUSY;                                  7840.000
                                                        S.P.M.INVALID                                               7841.000
0F5C B 4 0 0 1 0 0 0 00 4 00F 0 0 0 0 0 0 0 0  F5F              RSTPROTV, IF ALUZ *GO TO S.P.SET.ERR.A;             7842.000
0F5D 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0                   RESET(ENAUORD);                                     7843.000
0F5E 0 4 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0  F51              *GO TO S.P.CHECK.MAP.ERROR+1; CLEAR FALSE ERROR & EXIT  7844.000
                                                        S.P.SET.ERR.A                                               7845.000
0F5F 0 4 0 3 1 E F 0 02 0 0A1 0 0 0 0 0 0 0 0  F61              T=@0000000A, *HOP $+2; SET ERROR CODE A FOR MAP INVALID  7846.000
                                                        S.P.SET.ERR.9                                               7847.000
0F60 0 0 0 3 1 E F 0 02 0 090 0 0 0 0 0 0 0 0                   T=@000000009;  SET ERROR CODE 9 FOR MAP WRITE PROTECT CHECK  7848.000
0F61 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0                   RESET(ENAUORD);                                     7849.000
```

```
            S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    S E R I A L  P A N E L  C H F C K  M A P  E R R O R S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0F62 0 6 0 0 1 0 0 0 00 0 AD8 0 0 0 0 0 0 0 0  AD8        *GO TO S.P.CLEAR.ERROR;                                          7850.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL 32/75 CPU                                                    SERIAL PANEL CHECK MAP ERRORS
PC   T S M A B + D K  Y X PCH + B D M + R A D  ADDR

                                                                                                  7851.000
                                                *                                                 7852.000
                                                SIPU.IPU.TRAP                                     7853.000
                                                *                                                 7854.000
0F63 0 6 7 0 1 0 0 0 00 8 59C 0 0 0 0 0 0 0 0  59C      PCTOMAR,*LINK DECREMENT.PC;   UPDATE PC   7855.000
0F64 F 3 0 3 1 E F 0 02 0 100 0 0 0 0 0 0 0 0           T=@00000010,IF FALSE *JUMPZ;  ADVANCE PIPELINE 7856.000
0F65 0 6 7 0 1 0 0 0 00 0 596 0 0 0 0 0 0 0 0  596      *LINK UPDATE.CPU.STATUS;                  7857.000
0F66 0 0 0 0 1 0 0 0 00 0 D05 0 0 0 0 0 0 0 0           SET(IPU.HALT);                FLAG SIPU INTERNAL EVENT 7858.000
0F67 0 0 0 3 1 E F 0 02 0 F40 0 0 0 0 0 0 0 0           T=@000000F4;                              7859.000
0F68 0 0 0 2 0 A F 0 02 6 020 0 0 0 0 0 0 0 0           T=@00000200:T,SDEST;          FETCH IPU STOP TRAP 7860.000
0F69 0 6 0 0 1 0 0 0 00 0 580 0 0 0 0 0 0 0 0  580      *GOTO TRAP.MODE75;                        7861.000
                                                *                                                 7862.000
                                                TEST.IPU.CPU                                      7863.000
                                                *                                                 7864.000
0F6A 4 4 0 3 1 E F 0 02 2 00E 0 0 0 0 0 0 0 0  F6E      T=@0,IF IPU *HOP CONNECT.ERROR.TRAP;      7865.000
0F6B 2 4 0 0 1 E F 0 00 0 30D 0 0 0 0 0 0 0 0  F6D      T=S,IF MODE75 *GOTO $+2;                  7866.000
0F6C 0 6 0 0 1 0 0 0 00 0 569 0 0 0 0 0 0 0 0  569      *GOTO TRAP.MODE55;            CPU 55 MODE 7867.000
0F6D 0 6 0 0 1 0 0 0 00 0 580 0 0 0 0 0 0 0 0  580      *GOTO TRAP.MODE75;            CPU 75 MODE 7868.000
                                                *                                                 7869.000
                                                CONNECT.ERROR.TRAP                                7870.000
                                                *                                                 7871.000
0F6E 0 6 0 0 1 0 0 0 00 0 2D4 0 0 0 0 0 0 0 0  2D4      *GOTO IPU.ERROR.TRAP;                     7872.000
                                                *                                                 7873.000
                                                *                                                 7874.000
                                                EXTERNAL.IPU.TEST                                 7875.000
                                                *                                                 7876.000
0F6F 4 5 0 0 1 0 0 0 00 3 090 0 0 0 0 0 0 0 0  F90      IF IPU.START *GOTO MODE.HALT.CHECK;       7877.000
```

```
                S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    S E R I A L  P A N E L  C H E C K  M A P  E R R O R S
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0F70 0 6 0 0 1 0 0 0 00 0 004 0 0 0 0 0 0 0 0   4          *GOTO EVENT.POLL;                                                    7878.000

0F71 0 0 0 3 1 E F 0 02 0 100 0 0 0 0 0 0 0 0              T=@00000010;              FETCH IPU STATUS                           7879.000

0F72 4 6 7 0 1 0 0 0 00 2 596 0 0 0 0 0 0 0 0  596         IF IPU *LINK UPDATE.CPU.STATUS;     INSERT IPU STATUS            7880.000

0F73 0 0 0 0 4 0 F 0 00 0 F00 0 0 0 0 0 0 0 0              T=R(TRACE);               FETCH TRACE                                7881.000

0F74 0 0 0 4 0 D F 0 02 0 7F0 0 0 0 0 0 0 0 0              T=@7FFFFFFF&T;            CLEAR HALT BIT IN TRACE REG                7882.000

0F75 6 0 0 0 0 0 B 7 00 0 000 0 0 0 0 0 0 0 0              FR(TRACE)=T;              RESTORE TRACE                              7883.000

0F76 0 0 0 3 1 E F 0 02 0 E00 0 0 0 0 0 0 0 0              T=@000000E0;                                                         7884.000

0F77 0 0 0 2 0 A F 0 02 6 020 0 0 0 0 0 0 0 0              T=@00000200:T,SDEST;                                                 7885.000

0F78 0 6 0 0 1 0 0 0 00 0 0AE 0 0 0 0 0 0 0 0   AE         *GOTO IPU.TRAP;                                                      7886.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  79APR27
SEL  32/75  CPU                                                          75  MODE  CONTROL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR
                                                                                                                    7887.000
                                                    *                                                               7888.000
                                                    IPU.CHECK                                                       7889.000
                                                    *                                                               7890.000
0F79 4 5 0 0 4 0 0 4 00 2 004 0 0 0 0 0 0 0 0  F04          NOD=R(R),IF IPU *GOTO IPU.TEST.75;                      7891.000
0F7A 0 5 0 0 4 0 0 4 00 0 007 0 0 0 0 0 0 0 0  F07          NOD=R(R),*GOTO SET.55.EXIT;                             7892.000
                                                    *                                                               7893.000
                                                    SVC.CHECK                                                       7894.000
                                                    *                                                               7895.000
0F7B 4 4 0 3 1 E 1 0 02 C 180 0 0 0 0 0 0 0 0  F70          S=@00000018,IF MODE75S *HOP $+2;                        7896.000
0F7C 0 6 0 0 1 0 0 0 00 0 5CE 0 0 0 0 0 0 0 0  5CE          *GOTO UNDEF.55.5XX;       SVC INVALID IN 55 MODE        7897.000
0F7D 4 4 0 0 1 0 0 0 00 2 00F 0 0 0 0 0 0 0 0  F7F          IF IPU *GOTO $+2;                                       7898.000
0F7E 0 6 0 0 1 0 0 0 00 0 5CF 0 0 0 0 0 0 0 0  5CF          *GOTO UNDEF.55.5XX+1;     SVC IN CPU MODE               7899.000
0F7F 0 0 0 0 1 0 0 0 00 0 705 0 0 0 0 0 0 0 0               RESET(FLAG);                                            7900.000
0F80 0 0 0 3 1 E F 0 02 0 F80 0 0 0 0 0 0 0 0               T=@000000E8;                                            7901.000
0F81 0 0 0 2 0 A E 0 02 0 020 0 0 0 0 0 0 0 0               FULLMAR=@00000200:T;      FETCH SVC TRAP FOR IPU        7902.000
0F82 0 6 0 0 1 0 0 0 00 0 5D0 0 0 0 0 0 0 0 0  5D0          *GOTO SVC.1;              RETURN TO SVC PROCESSOR       7903.000
                                                    *                                                               7904.000
                                                    *                                                               7905.000
                                                    IPU.CALM.PROC                                                   7906.000
                                                    *                                                               7907.000
0F83 F 3 0 0 1 0 0 0 00 5 000 0 0 0 0 0 0 0 0               TOGRHF,IF FALSE *JUMPZ;   ADVANCE PIPELINE              7908.000
0F84 0 6 7 0 1 0 0 0 00 8 59C 0 0 0 0 0 0 0 0  59C          PCTOMAR,*LINK DECREMENT.PC; UPDATE PC                   7909.000
0F85 0 0 0 3 1 E F 0 02 0 100 0 0 0 0 0 0 0 0               T=@00000010;                 FETCH IPU STATUS BIT       7910.000
0F86 0 6 7 0 1 0 0 0 00 0 596 0 0 0 0 0 0 0 0  596          *LINK UPDATE.CPU.STATUS;                                7911.000
0F87 0 0 0 3 1 E F 0 02 0 F00 0 0 0 0 0 0 0 0               T=@000000F0;                                            7912.000
0F88 0 0 0 2 0 A F 0 02 6 020 0 0 0 0 0 0 0 0               T=@00000200:T,SDEST;         FETCH CALM TRAP            7913.000
```

```
            S Y S T E M S  M I C R O C O D E  A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    7 5  M O D E  C O N T R O L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0F89 0 6 0 0 1 0 0 0 00 0 580 0 0 0 0 0 0 0 0  580          *GOTO TRAP.MODE75;                                    7914.000

                                                *                                                                  7915.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU 75 MODE CONTROL

```
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

                                                                                                                    7916.000

                                                                                                                    7917.000
                                                *                                                                   7918.000
                                                *       THIS ROUTINE IS ENTERED WHEN THE IPU MODE HAS BEEN          7919.000
                                                *       ESTABLISHED AND THE 75 TRAPS ARE DISABLED (BIT 25 SET).     7920.000
                                                *       THE FLAG 'IPU.HALT' IS USED HERE TO DETERMINE WHEN          7921.000
                                                *       A RETURN SHOULD BE MADE TO THE COMMON TRAP HANDLER          7922.000
                                                *       ROUTINE.                                                    7923.000
                                                *                                                                   7924.000
                                                IPU.TRAP.FLAG.TEST                                                  7925.000
                                                *                                                                   7926.000
0F8A 4 4 0 0 1 0 0 0 00 D 00D 0 0 0 0 0 0 0 0  F8D      IF IPU.HALT *GOTO $+3;    CHECK FLAG                      7927.000
0F8B 0 0 0 0 1 0 0 0 00 0 005 0 0 0 0 0 0 0 0           SET(IPU.HALT);            FLAG FIRST ENTRY                7928.000
0F8C 0 6 0 0 1 0 0 0 00 0 2D4 0 0 0 0 0 0 0 0  2D4      *GOTO IPU.ERROR.TRAP;     ACTIVATE ERROR PROCESSOR        7929.000
0F8D 0 0 0 0 1 0 0 0 00 0 505 0 0 0 0 0 0 0 0           RESET(IPU.HALT);          CLEAR FLAG                      7930.000
0F8E 0 6 0 0 1 0 0 0 00 0 58F 0 0 0 0 0 0 0 0  58E      *GOTO SVC.TRAP.ENTRY.POINT;    RETURN TO HANDLER          7931.000
                                                *                                                                   7932.000
                                                (@F90)                                                              7933.000
                                                *                                                                   7934.000
                                                MODE.HALT.CHECK                                                     7935.000
                                                *                                                                   7936.000
0F90 4 5 0 0 1 0 0 0 00 2 071 0 0 0 0 0 0 0 0  F71      IF IPU *GOTO EXTERNAL.IPU.TEST+2;                           7937.000
0F91 0 6 0 0 1 0 0 0 13 0 004 0 0 0 0 0 0 0 0    4      RSTIPUFLG,*GOTO EVENT.POLL;                                 7938.000
                                                                                                                    7939.000
                                                *                                                                   7940.000
                                                ***                                                                 7941.000
                                                ***  NON-INTERRUPTABLE EXIT PROVIDES A NON-INTERRUPTABLE            7942.000
                                                ***    SEQUENCE FROM ONE INSTRUCTION TO THE NEXT IF                 7943.000
                                                ***    BIT 16 OF R(OFFSET) IS SET AND THE ENABLE INTERRUPT F/F IS   7944.000
                                                ***    RESET BY THE CALLING SUB-ROUTINE.                            7945.000
                                                ***                                                                7946.000

                                                                                                                    7947.000

                                                (@FB0)                                                              7948.000

                                                NON.INTERRUPTABLE.EXIT                                              7949.000

0FB0 9 4 0 0 1 0 0 0 00 0 5B2 0 0 0 0 0 0 0 0  FB2      IF %IONORESP:IOTIMEOUT *GO TO $+2;                          7950.000

0FB1 0 6 7 0 1 0 0 0 00 0 515 0 0 0 0 0 0 0 0  515      *LINK CLEAR.TIMEOUT;                                        7951.000

0FB2 6 0 0 0 4 0 0 4 19 0 084 0 0 0 0 0 0 0 0           NOD=FR(OFFSET), FETCHPC, RESET(HIREG);                      7952.000

0FB3 3 3 0 0 1 0 0 0 00 8 000 0 0 0 0 0 0 0 0           IF BMUX16 *JUMP7;  EXIT IF 55 MODE                          7953.000

0FB4 0 0 4 0 2 0 0 0 00 7 104 0 0 0 0 0 0 0 0           NOD=I0, I1TOI0, RESET(EXFF), DECODE(0);SET PRIMARY DECODE FOR I1 7954.000

0FB5 9 2 6 0 1 0 F 0 00 0 450 0 0 0 0 0 0 0 0           T=MAR, DECODE(#), IF %OPNORESP:OPTIMEOUT:OPMIUER:OPRNOPE *JUMP0; 7955.000

0FB6 0 6 0 0 1 0 0 0 00 0 001 0 0 0 0 0 0 0 0    1      *GO TO EXTG;  EXIT TO EXTERNAL GLOBAL SCHEDULING            7956.000

                                                COMPUTE.CPU.STATUS                                                  7957.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

```
SEL  32/75  CPU                                                    75  MODE  CONTROL
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0FB7 A 1 0 0 1 E F 0 00 0 300 0 0 0 0 0 0 0 0          T=S, IF %MODE75 *JUMPJ;                                    7958.000

0FB8 4 4 7 3 1 E F 0 02 C 00D 0 0 0 0 0 0 0 0  FBD     T=@00000000, IF MODE75S *LINK SET.75.M;                    7959.000

0FB9 4 4 7 0 1 0 0 0 00 A 00E 0 0 0 0 0 0 0 0  FBE     IF ENBL.AEXP *LINK SET.ENBL.AEXP;                          7960.000

0FBA 4 4 7 0 1 0 0 0 00 9 00F 0 0 0 0 0 0 0 0  FBF     IF ELSA *LINK SET.ELSA.M;                                  7961.000

0FBB 4 1 0 0 1 0 0 0 00 8 000 0 0 0 0 0 0 0 0          IF UNBLOCK *JUMPJ;                                         7962.000

0FBC 0 1 0 3 0 A F 0 02 0 200 0 0 0 0 0 0 0 0          T=@00000020:T, *JUMPJ; SET BLOCK MODE FLAG                 7963.000

                                                SET.75.M                                                          7964.000

0FBD 0 1 0 3 0 A F 0 02 0 010 0 0 0 0 0 0 0 0          T=@00000001:T, *JUMPJ;  SET 75 MODE FLAG                   7965.000

                                                SET.ENBL.AEXP                                                     7966.000

0FBE 0 1 0 3 0 A F 0 02 0 800 0 0 0 0 0 0 0 0          T=@00000080:T, *JUMPJ;  SET AEXP ENABLED FLAG              7967.000

                                                SET.ELSA.M                                                        7968.000

0FBF 0 1 0 3 0 A F 0 02 0 040 0 0 0 0 0 0 0 0          T=@00000004:T, *JUMPJ;  SET ELSA MODE FLAG                 7969.000

                                                                                                                  7970.000
                                                ***                                                               7971.000
                                                ***                                                               7972.000
                                                *** READ CPU STATUS INSTRUCTION  -RDSTS-, OP CODE =@00 , Q = 00   7973.000
                                                ***                                                               7974.000

                                                                                                                  7975.000

                                                RDSTS                                                             7976.000

0FC0 0 6 7 0 1 0 0 0 00 0 7F6 0 0 0 0 0 0 0 0  7F6     *LINK CHECK.NOT.55.ENPL; CHECK CPU IS NOT RESTRICTED TO 55 MODE  7977.000

0FC1 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0          S=SCRATCH(@91);  FETCH CPU STATUS WORD                     7978.000

0FC2 0 5 7 0 1 0 0 0 00 0 0B7 0 0 0 0 0 0 0 0  FB7     *LINK COMPUTE.CPU.STATUS;  GO COMPUTE MODE STATUS            7979.000

0FC3 0 0 0 0 0 0 8 0 00 0 B00 0 0 0 0 0 0 0 0          R(TEMP1)=T; SAVE MODE STATUS                               7980.000

0FC4 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0          T=S;                                                       7981.000

0FC5 0 0 0 7 0 D 1 0 00 0 400 0 0 0 0 0 0 0 0          S=@FFFFFF40&T; SAVE OLD ERROR STATUS & TRAP FLAG           7982.000

0FC6 0 0 0 7 0 D 0 0 00 0 000 0 0 0 0 0 0 0 0          NOD=@FFFFFF00&T; TEST OLD ERROR STATUS FOR ZERO            7983.000

0FC7 0 0 0 0 4 A F 0 00 0 B07 0 0 0 0 0 0 0 0          T=S:R(TEMP1), CLRS; COMBINE OLD ERROR STATUS AND NEW MODE STATUS  7984.000

0FC8 C 4 0 0 0 0 9 0 00 0 00A 0 0 0 0 0 0 0 0  FCA     R(R)=T, IF NALU7 *GO TO $+2;  SAVE CPU UPDATED STATUS WORD   7985.000

0FC9 0 0 0 0 1 E 1 0 02 0 080 0 0 0 0 0 0 0 0          S=@08000000; SET CC CODE FOR ERROR STATUS  =0               7986.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    7 5  M O D E  C O N T R O L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0FCA 0 0 0 3 0 D F 0 02 0 FF0 0 0 0 0 0 0 0 0          T=@000000FF&T;  CLEAR ERROR STATUS                                    7987.000

0FCB 0 0 0 1 0 0 4 0 02 A 910 0 0 0 0 0 0 0 0          SCRATCH(@91)=T;  STORE CPU STATUS WORD WITH ERROR STATUS =0           7988.000

0FCC A 4 0 0 1 E F 0 00 0 30E 0 0 0 0 0 0 0 0  FCE     T=S, IF %MODE75 *GO TO $+2;                                          7989.000

0FCD 0 0 0 0 0 A F 0 02 0 200 0 0 0 0 0 0 0 0          T=@20000000:T; SET CC CODE FOR 75 MODE                               7990.000

0FCE 4 5 0 0 1 0 0 0 00 8 0D0 0 0 0 0 0 0 0 0  FD0     IF UNBLOCK *GO TO $+2;                                               7991.000

0FCF 0 0 0 0 0 A F 0 02 0 100 0 0 0 0 0 0 0 0          T=@10000000:T; SET CC CODE FOR BLOCKED INTERRUPTS                    7992.000

0FD0 0 6 1 0 0 0 1 0 00 0 829 0 0 0 0 0 0 0 0  829     S=T, SETCC(#), *GO TO FETCH.RETURN;                                  7993.000

                                                                                                                            7994.000
                                                ** SET CPU MODE -SETCPU-, SECONDARY DECODE (0=08)                           7995.000
                                                **                                                                          7996.000
                                                **   ENTRY PARAMETERS :                                                     7997.000
                                                **     R(R) BIT 16-19 CONTAIN THE CPU MODE                                  7998.000
                                                **     BIT 19=1 75 MODE; =0 55 MODE                                         7999.000
                                                **     BIT 17 =1 MODE ELSA                                                  8000.000
                                                **     BIT 15 =1 CLEAR DISABLE TRAP FLAG                                    8001.000
                                                **     BIT 14 =1 CLEAR DISABLE BLOCK MODE TIMEOUT                           8002.000
                                                **                                                                          8003.000
                                                                                                                            8004.000
                                                                                                                            8005.000
                                                SETCPU                                                                      8006.000

0FD1 0 6 7 0 1 0 0 0 00 0 7F6 0 0 0 0 0 0 0 0  7F6     *LINK CHECK.NOT.55.ENBL; GO VERIFY CORRECT ENTRY PARAMETERS           8007.000

0FD2 0 0 0 0 4 0 0 4 00 0 206 0 0 0 0 0 0 0 0          NOD=R(R), RESET(ELSA); CLEAR ALL CPU MODES                           8008.000

                                                TEST.75                                                                     8009.000

0FD3 3 4 0 0 4 0 0 4 00 8 007 0 0 0 0 0 0 0 0  FD7     NOD=R(R), IF BMUX19 *GO TO SET.55.EXIT;                              8010.000

                                                *                                                                           8011.000

                                                IPU.TEST.75                                                                 8012.000

                                                *                                                                           8013.000

0FD4 0 0 0 0 4 0 0 4 00 0 806 0 0 0 0 0 0 0 0          NOD=R(R), SET(MODE75);                                               8014.000

                                                TEST.ELSA                                                                   8015.000

0FD5 3 4 0 0 4 0 0 4 00 9 00F 0 0 0 0 0 0 0 0  FDF     NOD=R(R), IF BMUX17 *GO TO SETCPU.EXIT;                              8016.000

0FD6 0 0 0 0 4 0 0 4 00 0 A06 0 0 0 0 0 0 0 0          NOD=R(R), SET(ELSA);                                                 8017.000

                                                SET.55.EXIT                                                                 8018.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  79APR27
S E L  32/75  C P U                                                          75  M O D E  C O N T R O L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0FD7 C 4 0 0 1 0 0 0 00 0 00F 0 0 0 0 0 0 0 0  FDF        IF NALUZ *GO TO SETCPU.EXIT; DO NOT SET 55 IF BITS 19=1 &14/15=0 8019.000
0FD8 4 4 0 0 1 0 0 0 00 8 00B 0 0 0 0 0 0 0 0  FD8        IF UNBLOCK *GO TO $+3;                                          8020.000
0FD9 0 0 0 1 1 E 1 0 02 0 080 0 0 0 0 0 0 0 0              S=UNDEF.75.FLG;                                                 8021.000
0FDA 0 6 0 0 1 0 0 0 00 0 577 0 0 0 0 0 0 0 0  577        *GO TO MACHINE.CHECK.TRAP;                                      8022.000
0FDB 1 4 0 0 1 0 0 0 00 0 F79 0 0 0 0 0 0 0 0  FD9        IF MAPMODE *HOP $-2;                                            8023.000
0FDC 0 0 0 0 1 0 0 0 00 0 F06 0 0 0 0 0 0 0 0              SET(DIS.BLK.TIMEOUT);                                           8024.000
0FDD 0 0 0 0 1 0 0 0 00 0 306 0 0 0 0 0 0 0 0              RESET(ENBL.AEXP);                                               8025.000
0FDE 0 0 0 0 1 0 0 0 00 0 006 0 0 0 0 0 0 0 0              RESET(MODE75);  GO TO 55 MODE                                   8026.000
                                                 SETCPU.EXIT                                                           8027.000
0FDF 0 5 7 0 1 0 0 0 00 0 087 0 0 0 0 0 0 0 0  FB7        *LINK COMPUTE.CPU.STATUS;                                       8028.000
0FE0 0 0 0 0 0 0 8 0 00 0 B00 0 0 0 0 0 0 0 0              R(TEMP1)=T;   SAVE MODE STATUS                                  8029.000
0FE1 0 0 0 1 1 0 1 0 02 A 910 0 0 0 0 0 0 0 0              S=SCRATCH(@91); FETCH CPU STATUS WORD                           8030.000
0FE2 0 0 0 0 1 E F 0 00 0 000 0 0 0 0 0 0 0 0              T=S;                                                            8031.000
0FE3 0 0 0 1 4 D 0 4 02 0 010 0 0 0 0 0 0 0 0              NOD=@00010000&R(R); TEST FOR ENABLE TRAP CODE                   8032.000
0FE4 0 0 0 7 0 D F 0 00 0 C00 0 0 0 0 0 0 0 0              T=@FFFFFFC0&T; GET ERROR STATUS AND DISABLE AEXP & TRAP FLAGS   8033.000
0FE5 B 4 0 4 0 A F 0 00 0 B07 0 0 0 0 0 0 0 0  FE7        T=R(TEMP1):T, IF ALUZ *GO TO $+2; COMBINE ERROR AND MODE STATUS 8034.000
0FE6 0 0 0 7 0 D F 0 00 0 BF0 0 0 0 0 0 0 0 0              T=@FFFFFFBF&T;  CLEAR DISABLE TRAPS                             8035.000
0FE7 0 0 0 1 4 D 0 4 02 0 020 0 0 0 0 0 0 0 0              NOD=@00020000&R(R); TEST FOR ENABLE BLOCK MODE TIMEOUT BIT      8036.000
0FE8 0 0 0 1 0 0 4 0 02 A 910 0 0 0 0 0 0 0 0              SCRATCH(@91)=T; STORE CPU FLAGS                                 8037.000
0FE9 C 0 0 0 1 0 0 0 00 0 706 0 0 0 0 0 0 0 0              IF NALUZ RESET(DIS.BLK.TIMEOUT); DISABLE BLOCK MODE TIMEOUTS    8038.000
0FEA 0 6 0 0 1 0 0 0 00 0 829 0 0 0 0 0 0 0 0  829        *GO TO FETCH.RETURN;                                            8039.000
                                                 SET.BLOCKED.M.FLAG                                                    8040.000
0FEB 6 0 0 2 4 A B 7 02 D 400 0 0 0 0 0 0 0 0              FR(TRACE)=@00004000:FR(TRACE), OTHERBANK; SET BLOCKED MODE FLAG 8041.000
0FEC 0 0 0 0 1 0 0 0 00 D 906 0 0 0 0 0 0 0 0              OTHERBANK, SET(UNBLOCK); REFRESH BLOCK TIMEOUT COUNT            8042.000
0FED 0 1 0 0 1 E F 0 00 0 106 0 0 0 0 0 0 0 0              T=S, RESET(UNBLOCK), *JUMPJ;                                    8043.000
                                                 EFFECTIVE.ADDR.INDIR                                                  8044.000
0FEE 0 0 0 0 1 0 0 0 1C 0 080 0 0 0 0 0 0 0 0              READ; FETCH INDIRECT WORD                                       8045.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
SEL  32/75  CPU                                                                  7 5   M O D E   C O N T R O L
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR

0FEF 0 0 0 0 0 0 8 0 00 0 D00 0 0 0 0 0 0 0 0          R(TFMP3)=T;  SAVE EFFECTIVE ADDRESS MASK                          8046.000

                                                  LOOP.EFFECTIVE.INDIR                                               8047.000

0FF0 0 0 0 3 1 D F 0 02 0 030 0 0 0 0 0 0 0 0          T=@00000003&MAR; SAVE C-BITS                                      8048.000

0FF1 0 0 0 4 3 3 7 1 00 6 000 0 0 0 0 0 0 0 0          MARTX=R(DIX)+DI, SDEST;                                           8049.000

0FF2 0 0 0 0 1 0 0 0 1C 0 080 0 0 0 0 0 0 0 0          READ; FETCH INDIRECT WORD OR OPERAND                              8050.000

0FF3 9 4 0 0 1 0 0 0 00 0 555 0 0 0 0 0 0 0 0  FF5     IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2; TEST FOR PREVIOUS ERRORS 051.000

0FF4 0 6 0 0 0 1 C 0 00 F 304 0 0 0 0 0 0 0 0  304     NU=%T(ZE), *GO TO CD.DUD; EXIT TO ERROR PROCESSING                8052.000

0FF5 2 4 0 0 1 0 0 0 00 0 C00 0 0 0 0 0 0 0 0  FF0     IF INDIR *GO TO LOOP.EFFECTIVE.INDIR;                             8053.000

0FF6 5 4 0 0 3 0 0 0 00 F 009 0 0 0 0 0 0 0 0  FF9     NOD=DI, IF %BYTF *GO TO MERGE.F.C.BITS;                           8054.000

0FF7 A 4 0 0 1 0 0 0 00 0 909 0 0 0 0 0 0 0 0  FF9     IF %NCTRZ *GO TO MERGE.F.C.BITS; BRANCH IF EXT INDEX              8055.000

0FF8 0 0 0 1 0 A F 0 02 0 080 0 0 0 0 0 0 0 0          T=@00080000:T; SET F BIT IN T (WITH C-BITS)                       8056.000
                                                                                                                     . . .
                                                  MERGE.F.C.BITS                                                     8057.000

0FF9 0 0 0 0 0 0 C 0 00 F 000 0 0 0 0 0 0 0 0          NU=T(ZE); CLEAR ERROR FLAGS                                       8058.000

0FFA 9 4 0 0 0 A 1 0 00 0 55C 0 0 0 0 0 0 0 0  FFC     S=S:T, IF %OPNORESP:OPTIMEOUT:OPRNOPE *HOP $+2;                   8059.000

0FFB 0 6 7 0 1 0 0 0 00 0 2FA 0 0 0 0 0 0 0 0  2FA     *LINK CLEAR.MEM.ERROR; GO PURGE MEMORY ERROR                      8060.000

0FFC 0 1 0 0 4 0 F 0 00 0 D00 0 0 0 0 0 0 0 0          T=R(TEMP3), *JUMPJ; RESTORE MASK AND EXIT                         8061.000

                                                                                                                     8062.000
                                                  **                                                                 8063.000
                                                  ** UNDEFIFED CPU TAGS                                              8064.000
                                                  **                                                                 8065.000

                                                                                                                     8066.000

                                                  (@FFE)                                                             8067.000

                                                                                                                     8068.000

0FFE 0 4 0 0 2 0 0 0 00 0 00E 0 0 0 0 0 0 0 0  FFE     NOD=I0, *GO TO $;       DUMMY ERROR TRAP                          8069.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
                                                                D   R O M   C O N T E N T S
S E L   3 2 / 7 5   C P U
PC   T S M A B + D R  Y X PCH + B D M + B A D  ADDR                                                          8070.000


                                                                                                            8071.000
*************************************************************************                                   8072.000
*** NAME                 : DROM (DROM)                                  ***                                 8073.000
*** PART NUMBER/REV      : 531-322701/007                               ***                                 8074.000
*** LOCATIONS            : COPPER C BOARD E07,F07,B08,B09,E06,          ***                                 8075.000
***                      : AND C09 (MOST SIGNIFICANT NIBBLE TO          ***                                 8076.000
***                      : LEAST SIGNIFICANT NIBBLE)                    ***                                 8077.000
*** DATE                 : FEBUARY 05,1979                              ***                                 8078.000
*** OBJECT FILE          : D7502                                        ***                                 8079.000
*************************************************************************                                   8080.000


                                                                                                            8081.000
        $USF    DRDEF                                                                                       8082.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU   SC    CC   PAGE  ADDR

                                                                                                                8083.000
                                    ***                                                                         8084.000
                                    *** D-ROM, PAGES 0-3 PRIMARY DECODE OF FULLWORD ***                         8085.000
                                    *** AND LEFT HALFWORD INSTRUCTIONS ***                                      8086.000
                                    *** (DECODE I1 OR I0 BITS 00-05) ***                                        8087.000
                                    *** D-ROM.ADDR=IX(00,01)*16+IX(02,03,04,05) ***                             8088.000
                                    ***                                                                         8089.000
0000   86  4028                 86   805   (0)     NP=@86,   CTL       ;OP CODE=00 -HALT,WAIT,NOP,LCS,ES,RND,BEI,  8090.000
                                    *                                                UEI,EAE,RDSTS,RWCS,WWCS,SEA,  8091.000
                                    *                                                DAE,CEA-                      8092.000
0001   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=04 -ANR-                           8093.000
0002   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=08 -ORR,ORRM-                      8094.000
0003   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=0C -EOR,EORM-                      8095.000
0004   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=10 -CAR-                           8096.000
0005   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=14 -CMR-                           8097.000
0006   49  4178                 49   82F         NP=@49,   BITREGS   ;OP CODE=18 -SBR-                           8098.000
0007   49  4178                 49   82F         NP=@49,   BITREGS   ;OP CODE=1C -ZBR-                           8099.000
0008   49  4178                 49   82F         NP=@49,   BITREGS   ;OP CODE=20 -ABR-                           8100.000
0009   49  4178                 49   82F         NP=@49,   BITREGS   ;OP CODE=24 -TBR-                           8101.000
000A   00  4960                  0   92C         X,        TRSW      ;OP CODE=28 -TRSW-                          8102.000
000B   96  4100                 96   820         NP=@96,   REGREG    ;OP CODE=2C -TRR,TRC,TRN,XCR,LMAP,RDSTS,    8103.000
                                    *                                                TRRM,SETCPU,TMAPR,TRCM,TRNM,  8104.000
                                    *                                                XCRM,TRSC,TSCR-               8105.000
000C   00  4C70                  0   98E         X,        CALM      ;OP CODE=30 -CALM-                          8106.000
000D   49  2F00                 49   5E0         NP=@49,   LA        ;OP CODE=34 -LA-                            8107.000
000E   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=38 -ADR,ADRM-                      8108.000
000F   BC  4100                 BC   820         NP=@BC,   REGREG    ;OP CODE=3C -SUR,SURM-                      8109.000
0010   49  33B0                 49   676   (@10)   NP=@49,   MPRDVR    ;OP CODE=40 -MPR-                       8110.000
0011   49  33B0                 49   676         NP=@49,   MPRDVR    ;OP CODE=44 -DVR-                           8111.000
0012   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8112.000
0013   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8113.000
0014   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8114.000
0015   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8115.000
0016   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8116.000
0017   00  4D40                  0   9A8         X,        UNDEF2    ;                                           8117.000
0018   00  42A0                  0   854         X,        NOR       ;OP CODE=60 -NOR-                           8118.000
0019   00  42E8                  0   85D         X,        NORD      ;OP CODE=64 -NORD-                          8119.000
001A   49  4530                 49   8A6         NP=@49,   SCZ       ;OP CODE=68 -SCZ-                           8120.000
001B   DD  4200                 DD   840         NP=@DD,   SHIFTS    ;OP CODE=6C -SRA,SLA-                       8121.000
001C   DD  4200                 DD   840         NP=@DD,   SHIFTS    ;OP CODE=70 -SRL,SLL-                       8122.000
001D   DD  4200                 DD   840         NP=@DD,   SHIFTS    ;OP CODE=74 -SRC,SLC-                       8123.000
001E   DD  4230                 DD   846         NP=@DD,   SHIFTD    ;OP CODE=78 -SRAD,SLAD-                     8124.000
001F   DD  4230                 DD   846         NP=@DD,   SHIFTD    ;OP CODE=7C -SRLD,SLLD-                     8125.000
0020   00  5000                  0   A00   (@20)   X,        LEAP      ;OP CODE=80 -LEAR-                          8126.000
0021   49  43B8                 49   877         NP=@49,   MRLOAD    ;OP CODE=84 -ANMW,ANMH,ANMD,ANMB-           8127.000
0022   49  43B8                 49   877         NP=@49,   MRLOAD    ;OP CODE=88 -ORMW,ORMH,ORMD,ORMB-           8128.000
0023   49  43B8                 49   877         NP=@49,   MRLOAD    ;OP CODE=8C -EOMW,EOMH,EOMD,EOMB-           8129.000
0024   49  43B8                 49   877         NP=@49,   MRLOAD    ;OP CODE=90 -CAMW,CAMH,CAMD,CAMB-           8130.000
0025   49  43B8                 49   877         NP=@49,   MRLOAD    ;OP CODE=94 -CMMW,CMMH,CMMD,CMMB-           8131.000
0026   49  47D8                 49   8FB         NP=@49,   BIT.MEM   ;OP CODE=98 -SBM-                           8132.000
0027   49  47D8                 49   8FB         NP=@49,   BIT.MEM   ;OP CODE=9C -ZBM-                           8133.000
0028   49  47D8                 49   8FB         NP=@49,   BIT.MEM   ;OP CODE=A0 -ABM-                           8134.000
0029   49  4850                 49   90A         NP=@49,   BIT.TBM   ;OP CODE=A4 -TBM-                           8135.000
002A   00  4B78                  0   96F         X,        EXM       ;OP CODE=A8 -EXM,EXMR-                      8136.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

S E L 3 2 / 7 5 C P U D R O M C O N T E N T S

```
PC   D(1) D(2)     ALU  SC   CC   PAGE ADDR
002B 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=AC -LW,LH,LD,LB-                    8137.000
002C 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=B0 -LMW,LMH,LMD,LMB-                8138.000
002D 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=B4 -LNW,LNH,LND,LNB-                8139.000
002E 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=B8 -ADMW,ADMH,ADMD,ADMB-            8140.000
002F 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=BC -SUMW,SUMH,SUMD,SUMB-            8141.000
0030 49   4388                   49   877   (@30)  NP=@49,    MRLOAD     ;OP CODE=C0 -MPMW,MPMH,MPMB-                 8142.000
0031 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=C4 -DVMW,DVMH,DVMB-                 8143.000
0032 8E   41D0                   8E   83A          NP=@8E ,   IMMED      ;OP CODE=C8 -LI,ADI,SUI,MPI,DVI,CI,SVC-       8144.000
                                                *                                  -EXR,EXRR,SEM,LEM,CEMA-       8145.000
0033 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=CC -LF-                             8146.000
0034 49   2F00                   49   5E0          NP=@49,    LEA        ;OP CODE=D0 -LEA-                            8147.000
0035 49   4698                   49   8D3          NP=@49,    MRSTORE    ;OP CODE=D4 -STW,STH,STD,STB-                8148.000
0036 49   4690                   49   8D2          NP=@49,    STM        ;OP CODE=D8 -STMW,STMH,STMD,STMB-            8149.000
0037 49   4698                   49   8D3          NP=@49,    MRSTORE    ;OP CODE=DC -STF-                            8150.000
0038 F1   6700                   F1   CE0        NP=@F1,ADD.SUB.FLT.PT ;OP CODE=E0 -SUFW,SUFD,ADFW,ADFD-            8151.000
0039 F9   6700                   F9   CE0        NP=@F9,MPY.DVD.FLT.PT ;OP CODE=E4 -DVFW,DFVD,MPFW,MPFD-            8152.000
003A 49   4388                   49   877          NP=@49,    MRLOAD     ;OP CODE=E8 -ARMW,ARMH,ARMD,ARMB-            8153.000
003B C7   4988                   C7   931          NP=@C7,    BCT        ;OP CODE=EC -BU,BCT-                         8154.000
003C C7   4940                   C7   928          NP=@C7,    BCF        ;OP CODE=F0 -BFT,BCF-                        8155.000
003D E0   4A20                   E0   944          NP=@E0,    BINC       ;OP CODE=F4 -BIB,BIH,BIW,BID-                8156.000
003E CF   4880                   CF   916          NP=@CF,    Q3E        ;OP CODE=F8 -ZMW,ZMH,ZMD,ZMB,BL,BRI,LPSD,     8157.000
                                                *                                  JWCS,LPSDCM,TRP,TPR-          8158.000
003F D3   4000                   D3   800          NP=@D3, INTR.CTL.IO   ;OP CODE=FC -EI,DI,RI,AI,DAI,TD,CD,SIO,      8159.000
                                                *                                  TIO,HIO,HCHNL,FRIO,ECWCS,     8160.000
                                                *                                  WCWCS,ECI,DCI,ACI,DACI-       8161.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU   SC    CC    PAGE  ADDR
                                                                                                                    8162.000
                                                                                                                    8163.000
                                                *** D-ROM, PAGES 4-7 PRIMARY DECODE OF RIGHT HALFWORD ***           8164.000
                                                *** INSTRUCTIONS (DECODE I1 BITS 16-21) ***                         8165.000
                                                *** D-ROM.ADDR=40+I1(16,17)*16+I1(18,19,20,21) ***                  8166.000
                                                ***                                                                 8167.000
0040   86   4028                      86    805  (@40)  NP=@86,   CTL       ;OP CODE=00 -HALT,WAIT,NOP,LCS,ES,RND,BEI,  8168.000
                                                *                                      UEI,EAE,RDSTS,RWCS,WWCS,SEA  8169.000
                                                *                                      DAE,CEA-                     8170.000
0041   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=04 -ANR-                       8171.000
0042   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=08 -ORR,ORRM-                  8172.000
0043   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=0C -EOR,EORM-                  8173.000
0044   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=10 -CAR-                       8174.000
0045   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=14 -CMR-                       8175.000
0046   49   4178                      49    82F        NP=@49,   BITREGS   ;OP CODE=18 -SBR-                       8176.000
0047   49   4178                      49    82F        NP=@49,   BITREGS   ;OP CODE=1C -ZBR-                       8177.000
0048   49   4178                      49    82F        NP=@49,   BITREGS   ;OP CODE=20 -ABR-                       8178.000
0049   49   4178                      49    82F        NP=@49,   BITREGS   ;OP CODE=24 -TBR-                       8179.000
004A   00   4960                       0    92C        X,        TRSW      ;OP CODE=28 -TRSW-                      8180.000
004B   96   4100                      96    820        NP=@96,   REGREG    ;OP CODE=2C -TRR,TRC,TRN,XCR,RDSTS,TRRM,    8181.000
                                                *                                      SETCPU,TMAPR,TRCM,TRNM,XCRM,  8182.000
                                                *                                      TRSC,TSCR-                   8183.000
004C   00   4C70                       0    98E        X,        CALM      ;OP CODE=30 -CALM-                      8184.000
004D   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8185.000
004E   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=38 -ADR,ADRM-                  8186.000
004F   BC   4100                      BC    820        NP=@BC,   REGREG    ;OP CODE=3C -SUR,SURM-                  8187.000
0050   49   33B0                      49    676  (@50)  NP=@49,   MPRDVR    ;OP CODE=40 -MPR-                       8188.000
0051   49   33B0                      49    676        NP=@49,   MPRDVR    ;OP CODE=44 -DVR-                       8189.000
0052   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8190.000
0053   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8191.000
0054   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8192.000
0055   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8193.000
0056   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8194.000
0057   00   4D30                       0    9A6        X,        UNDEF1    ;                                       8195.000
0058   00   42A0                       0    854        X,        NOR       ;OP CODE=60 -NOR-                       8196.000
0059   00   42E8                       0    85D        X,        NORD      ;OP CODE=64 -NORD-                      8197.000
005A   49   4530                      49    8A6        NP=@49,   SCZ       ;OP CODE=68 -SCZ-                       8198.000
005B   DD   4200                      DD    840        NP=@DD,   SHIFTS    ;OP CODE=6C -SRA,SLA-                   8199.000
005C   DD   4200                      DD    840        NP=@DD,   SHIFTS    ;OP CODE=70 -SRL,SLL-                   8200.000
005D   DD   4200                      DD    840        NP=@DD,   SHIFTS    ;OP CODE=74 -SRC,SLC-                   8201.000
005E   DD   4230                      DD    846        NP=@DD,   SHIFTD    ;OP CODE=78 -SRAD,SLAD-                 8202.000
005F   DD   4230                      DD    846        NP=@DD,   SHIFTD    ;OP CODE=7C -SRLD,SLLD-                 8203.000
0060   00   4D30                            9A6  (@60)              UNDEF1    ;                                       8204.000
0061   00   4D30                            9A6                     UNDEF1    ;                                       8205.000
0062   00   4D30                            9A6                     UNDEF1    ;                                       8206.000
0063   00   4D30                            9A6                     UNDEF1    ;                                       8207.000
0064   00   4D30                            9A6                     UNDEF1    ;                                       8208.000
0065   00   4D30                            9A6                     UNDEF1    ;                                       8209.000
0066   00   4D30                            9A6                     UNDEF1    ;                                       8210.000
0067   00   4D30                            9A6                     UNDEF1    ;                                       8211.000
0068   00   4D30                            9A6                     UNDEF1    ;                                       8212.000
0069   00   4D30                            9A6                     UNDEF1    ;                                       8213.000
006A   00   4D30                            9A6                     UNDEF1    ;                                       8214.000
006B   00   4D30                            9A6                     UNDEF1    ;                                       8215.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU D ROM CONTENTS

PC	D(1)	D(2)	ALU	SC	CC	PAGE	ADDR				
006C	00	4D30					9A6		UNDEF1	;	8216.000
006D	00	4D30					9A6		UNDEF1	;	8217.000
006E	00	4D30					9A6		UNDEF1	;	8218.000
006F	00	4D30					9A6		UNDEF1	;	8219.000
0070	00	4D30					9A6	(@70)	UNDEF1	;	8220.000
0071	00	4D30					9A6		UNDEF1	;	8221.000
0072	00	4D30					9A6		UNDEF1	;	8222.000
0073	00	4D30					9A6		UNDEF1	;	8223.000
0074	00	4D30					9A6		UNDEF1	;	8224.000
0075	00	4D30					9A6		UNDEF1	;	8225.000
0076	00	4D30					9A6		UNDEF1	;	8226.000
0077	00	4D30					9A6		UNDEF1	;	8227.000
0078	00	4D30					9A6		UNDEF1	;	8228.000
0079	00	4D30					9A6		UNDEF1	;	8229.000
007A	00	4D30					9A6		UNDEF1	;	8230.000
007B	00	4D30					9A6		UNDEF1	;	8231.000
007C	00	4D30					9A6		UNDEF1	;	8232.000
007D	00	4D30					9A6		UNDEF1	;	8233.000
007E	00	4D30					9A6		UNDEF1	;	8234.000
007F	00	4D30					9A6		UNDEF1	;	8235.000

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU   SC    CC   PAGE  ADDR
                                                                                                  8236.000
                                                                                                  8237.000
                                                                                                  8238.000
                                                                                                  8239.000
                                                                                                  8240.000
                                                                                                  8241.000
                                                                                                  8242.000
                                                                                                  8243.000
0080  00  4D48                          0    9A9     (@80)    X,        UNDEF     ;              8244.000
0081  00  4D48                          0    9A9              X,        UNDEF     ;              8245.000
0082  00  4D48                          0    9A9              X,        UNDEF     ;              8246.000
0083  00  4D48                          0    9A9              X,        UNDEF     ;              8247.000
0084  00  4D48                          0    9A9              X,        UNDEF     ;              8248.000
0085  00  4D48                          0    9A9              X,        UNDEF     ;              8249.000
0086  6C  41B0      D             4          836              -A.OR.B,  BIT,  SBRZBR;  -SBR-     8250.000
0087  0C  41B0      1             4          836              A.AND.B,  BIT,  SBRZBR;  -ZBR-     8251.000
0088  00  4188                    0          831              AL,       ABR       ;  -ABR-     8252.000
0089  04  4190                    4          832              BIT,      TBR       ;  -TBR-     8253.000
008A  00  4D48                          0    9A9              X,        UNDEF     ;              8254.000
008B  00  4D48                          0    9A9              X,        UNDEF     ;              8255.000
008C  00  4D48                          0    9A9              X,        UNDEF     ;              8256.000
008D  00  2F10                          0    5E2              X,        LA1       ;  -LA-      8257.000
008E  00  4D48                          0    9A9              X,        UNDEF     ;              8258.000
008F  00  4D48                          0    9A9              X,        UNDEF     ;              8259.000
                                                                                                  8260.000
                                                                                                  8261.000
                                                                                                  8262.000
                                                                                                  8263.000
                                                                                                  8264.000
                                                                                                  8265.000
                                                                                                  8266.000
0090  00  34F0                               69E     (@90)             MPR       ;  -MPR-     8267.000
0091  00  3420                               684                       DVR       ;  -DVR-     8268.000
0092  00  4D48                               9A9                       UNDEF     ;              8269.000
0093  00  4D48                               9A9                       UNDEF     ;              8270.000
0094  00  4D48                               9A9                       UNDEF     ;              8271.000
0095  00  4D48                               9A9                       UNDEF     ;              8272.000
0096  00  4D48                               9A9                       UNDEF     ;              8273.000
0097  00  4D48                               9A9                       UNDEF     ;              8274.000
0098  00  4D48                               9A9                       UNDEF     ;              8275.000
0099  00  4D48                               9A9                       UNDEF     ;              8276.000
009A  00  4D48                               9A9                       UNDEF     ;              8277.000
009B  00  4D48                               9A9                       UNDEF     ;              8278.000
009C  00  4D48                               9A9                       UNDEF     ;              8279.000
009D  00  4D48                               9A9                       UNDEF     ;              8280.000
009E  00  4D48                               9A9                       UNDEF     ;              8281.000
009F  00  4D48                               9A9                       UNDEF     ;              8282.000
                                                                                                  8283.000
                                                                                                  8284.000
                                                                                                  8285.000
                                                                                                  8286.000
                                                                                                  8287.000
                                                                                                  8288.000
                                                                                                  8289.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU   SC    CC   PAGE  ADDR
                                                                                                  8290.000
                                                ***                                               8290.000
00A0 00   4D48                          0   9A9  (@A0)  X,          UNDEF       ;  -LEAR-        8291.000
00A1 08   4480      1             0          890         A.AND.B,    AL,  ANM    ;  -ANMW,ANMH,ANMD,ANMB- 8292.000
00A2 20   4438      4             0          887         A.OR.B,     AL,  MRLOG  ;  -ORMW,ORMH,ORMD,ORMB- 8293.000
00A3 30   4438      6             0          887         A.EOR.B,    AL,  MRLOG  ;  -EOMW,EOMH,EOMD,EOMB- 8294.000
00A4 CF   44C0     19             7          898         A.MINUS.B,  C,   CAM    ;  -CAMW,CAMH,CAMD,CAMB- 8295.000
00A5 32   44F8      6             2          89F         A.EOR.B,    E,   CMM    ;  -CMMW,CMMH,CMMD,CMMB- 8296.000
00A6 24   4828      4             4          905         A.OR.B,     BIT, SRM    ;  -SRM-         8297.000
00A7 3C   4828      7             4          905         -A.AND.B,   BIT, ZRM    ;  -ZRM-         8298.000
00A8 B0   4840     16             0          908         A.PLUS.B,   AL,  ARM    ;  -ARM-         8299.000
00A9 0C   48A0      1             4          914         A.AND.B,    BIT, TRM    ;  -TRM-         8300.000
00AA 00   4D48                          0   9A9         X,          UNDEF       ;                8301.000
00AB 28   4408      5             0          881         B,          AL,  MRLAS  ;  -LW,LH,LD,LB- 8302.000
00AC 08   4470      1             0          88E         A.AND.B,    AL,  LM     ;  -LMW,LMH,LMD,LMB- 8303.000
00AD C8   4450     19             0          88A         A.MINUS.B,  AL,  LM     ;  -LNW,LNH,LND,LNB- 8304.000
00AE B0   4408     16             0          881         A.PLUS.B,   AL,  MRLAS  ;  -ADMW,ADMH,ADMD,ADMB- 8305.000
00AF C8   4408     19             0          881         A.MINUS.B,  AL,  MRLAS  ;  -SUMW,SUMH,SUMD,SUMB- 8306.000
                                                *** D-ROM, PAGE B  SECONDARY DECODE FOR ***        8307.000
                                                *** MEMORY REFERENCE INSTRUCTIONS ***              8308.000
                                                *** OP CODES = C0, C4, CC, D4, DA, DC, E8, EC, AND F0 ***  8309.000
                                                *** Q = 30, 31, 33, 34, 35, 37, 3A, 3B, AND 3C ***  8310.000
                                                *** (DECODE I0 BITS 0-1 & 2-5) ***                 8311.000
                                                *** D-ROM.ADDR=080+I0(00,01)*16+I0(02,03,04,05) ***  8312.000
                                                ***                                               8313.000
00B0 00   33C8                          0   679  (@B0)  X,          MPM         ;  -MPMW,MPMH,MPMB- 8314.000
00B1 00   33F8                          0   67F         X,          DVM         ;  -DVMW,DVMH,DVMB- 8315.000
00B2 00   4D48                          0   9A9         X,          UNDEF       ;                8316.000
00B3 00   4570                              8AE                     LF          ;  -LF-          8317.000
00B4 00   2F28                          0   5E5         X,          LEA1        ;  -LFA-         8318.000
00B5 28   46E8      5                       8DD         B,          ST          ;  -STW,STH,STD,STB- 8319.000
00B6 08   46E8      1                       8DD         A.AND.B,    ST          ;  -STMW,STMH,STMD,STMB- 8320.000
00B7 28   4710      5                       8E2         B,          STF         ;  -STF-         8321.000
00B8 00   4D48                          0   9A9         X,          UNDEF       ;                8322.000
00B9 00   4D48                          0   9A9         X,          UNDEF       ;                8323.000
00BA B0   4768     16             0          8ED         A.PLUS.B,   AL,  ARM    ;  -ARMW,ARMH,ARMD,ARMB- 8324.000
00BB 00   4D48                              9A9                     UNDEF       ;                8325.000
00BC 00   4D48                              9A9                     UNDEF       ;                8326.000
00BD 00   4D48                              9A9                     UNDEF       ;                8327.000
00BE 00   4D48                              9A9                     UNDEF       ;                8328.000
00BF 00   4D48                              9A9                     UNDEF       ;                8329.000
                                                ***                                               8330.000
                                                *** D-ROM, PAGE F SECONDARY DECODE FOR ***         8331.000
                                                *** FIRMWARE FLOATING POINT , SINGLE AND DOUBLE PRECISION ***  8332.000
                                                *** OP CODES = E0 & E4 , Q = 38 & 39 ***           8333.000
                                                *** (DECODE I0 BITS 12 & 3-5) ***                  8334.000
                                                *** D-ROM.ADDR=0F0+I0(12,03,04,05) ***             8335.000
                                                                                                  8336.000
                                                ***                                               8337.000
                                                *** D-ROM, PAGE F, THIRD LEVEL DECODE FOR ***      8338.000
                                                *** LPSD  AND LPSDCM INSTRUCTIONS ***              8339.000
                                                *** OP CODES = F98 & FA8, Q=3E ***                 8340.000
                                                *** (DECODE I0 BITS 0-1 & 5-8) ***                 8341.000
                                                *** D-ROM.ADDR=0F0+I0(05,06,07,08) ***             8342.000
```

```
PC     D(1) D(2)      ALU    SC     CC    PAGE  ADDR
                                                     ***

                                                     ***
                                                     *** SPECIAL DECODE FOR THE WCS DIAGNOSTIC SOFTWARE ***
                                                     *** TEST JUMPD TO A WCS ADDRESS FUNCTION ***
                                                     *** THE JUMPD TEST IS ACCESSED BY A MICROCODE 'DECODE(#)', ***
                                                     ***  '*JUMPD' SEQUENCE. ***
                                                     ***
                                                     ***
00F0   C8  3180      19            0           630     (@0F0)   A.MINUS.B, AL,   SUF  ;   -SUFW,SUFD-
00F1   00  3628                    0           6C5              AL ,      DVF         ;   -DVFW,DVFD-
00F2   00  4D48                                9A9                        UNDEF       ;
00F3   00  4E20                          0     9C4              X,        LPSD.EXIT   ;   -LPSD-
00F4   00  4D48                                9A9                        UNDEF       ;
00F5   00  4E28                          0     9C5              X,        LPSDCM.EXIT;    -LPSDCM-
00F6   00  4D48                                9A9                        UNDEF       ;
00F7   00  4D48                                9A9                        UNDEF       ;
00F8   B0  3180      16            0           630              A.PLUS.B, AL, ADF     ;   -ADFW,ADFD-
00F9   00  32C8                    0           659              AL,       MPF         ;   -MPFW,MPFD-
00FA   00  4D48                                9A9                        UNDEF       ;
00FB   00  4D48                                9A9                        UNDEF       ;
00FC   00  4D48                                9A9                        UNDEF       ;
00FD   00  4D48                                9A9                        UNDEF       ;
00FE   00  BFC0                          0    17F8              X,        @17F8       ;   -USED BY THE WCS TEST PROGRAM-
00FF   00  4D48                                9A9                        UNDEF       ;
                                                     ***
                                                     *** D-ROM, PAGE 10 SECONDARY DECODE FOR ***
                                                     *** CONTROL CLASS INSTRUCTIONS ***
                                                     *** OP CODE = 00 , Q = 00 ***
                                                     *** (DECODE IO BITS 12-15) ***
                                                     *** D-ROM.ADDR=100+IO(12,13,14,15) ***
                                                     ***
0100   00  4F40                          0     9E8     (@100)   X,        HALT        ;   -HALT-
0101   00  4FA8                          0     9F5              X,        WAIT        ;   -WAIT-
0102   00  4148                          0     829              X,        NOP         ;   -NOP-
0103   00  4038                    0           807              AL,       LCS         ;   -LCS-
0104   00  4078                          0     80F              X,        ES          ;   -ES-
0105   00  4090                          0     812              X,        RND         ;   -RND-
0106   00  3F48                          0     7E9              X,        BEI         ;   -BEI-
0107   00  3F30                          0     7E6              X,        UEI         ;   -UEI-
0108   00  2F70                          0     5EE              X,        EAE         ;   -EAE-
0109   05  7F00                    5           FC0              S,        RDSTS       ;   -RDSTS-
010A   00  1010                          0     202              X,        SIPU        ;   -SIPU-
010B   05  50A8                    5           A15              S,        RWCS        ;   -RWCS-
010C   05  5118                    5           A23              S,        WWCS        ;   -WWCS-
010D   00  4920                          0     924              X,        SEA         ;   -SEA-
010E   00  2F88                          0     5F1              X,        DAE         ;   -DAE-
010F   00  4928                          0     925              X,        CEA         ;   -CEA-
                                                     ***
                                                     *** D-ROM, PAGE 11 SECONDARY DECODE FOR ***
                                                     *** IMMEDIATE CLASS INSTRUCTIONS ***
                                                     *** OP CODE = C8 , Q = 32 ***
                                                     *** (DECODE IO BITS 12-15) ***
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU  SC   CC   PAGE ADDR
                                                                                                                    
                                    *** D-ROM.ADDR=110+IO(12,13,14,15) ***                                          8396.000
                                    ***                                                                             8397.000
0110 28   41E0      5          0         83C  (@110)  B, AL,    LASI       ;   -LI-                                8398.000
0111 B0   41E0     16          0         83C          A.PLUS.B, AL,  LASI  ;   -ADI-                               8399.000
0112 C8   41E0     19          0         83C          A.MINUS.B, AL, LASI  ;   -SUI-                               8400.000
0113 00   33E0                          0  67C          X,         MPI      ;   -MPI-                               8401.000
0114 00   3410                          0  682          X,         DVI      ;   -DVI-                               8402.000
0115 00   41F0                          0  83E          X,         CI       ;   -CI-                                8403.000
0116 00   2E60                          0  5CC          X,         SVC      ;   -SVC-                               8404.000
0117 00   4B30                          0  966          X,         EXR      ;   -EXR,EXRR-                          8405.000
0118 00   4D48                          0  9A9          X,         UNDEF    ;   -SEM-                               8406.000
0119 00   4D48                          0  9A9          X,         UNDEF    ;   -LEM-                               8407.000
011A 00   4D48                          0  9A9          X,         UNDEF    ;   -CEMA-                              8408.000
011B 00   4D48                          0  9A9          X,         UNDEF    ;                                       8409.000
011C 00   4D48                          0  9A9          X,         UNDEF    ;                                       8410.000
011D 00   4D48                          0  9A9          X,         UNDEF    ;                                       8411.000
011E 00   4D48                          0  9A9          X,         UNDEF    ;                                       8412.000
011F 00   4D48                          0  9A9          X,         UNDEF    ;                                       8413.000
                                    ***                                                                             8414.000
                                    *** D-ROM, PAGE 12 SECONDARY DECODE FOR ***                                      8415.000
                                    *** REGISTER TO REGISTER TRANSFER INSTRUCTIONS ***                               8416.000
                                    *** OP CODE = 2C , Q = 0B ***                                                   8417.000
                                    *** (DECODE IO BITS 12-15) ***                                                  8418.000
                                    *** D-ROM.ADDR=120+IO(12,13,14,15) ***                                          8419.000
                                    ***                                                                             8420.000
0120 28   4110      5                   822  (@120)  B,         RR1      ;   -TRR-                               8421.000
0121 00   4D48                          0  9A9          X,         UNDEF    ;                                       8422.000
0122 00   4D48                          0  9A9          X,         UNDEF    ;                                       8423.000
0123 50   4110      A          0         822          -B, AL,    RR1      ;   -TRC-                               8424.000
0124 C8   4118     19          0         823          A.MINUS.B, AL,  TRN ;   -TRN-                               8425.000
0125 00   40A8                0         815          AL,        XCR      ;   -XCR-                               8426.000
0126 00   4D48                          0  9A9          X,         UNDEF    ;                                       8427.000
0127 00   2080                          0  410          X,         LMAP     ;   -LMAP-                              8428.000
0128 28   4130      5          0         826          B, AL,     RRM      ;   -TRRM-                              8429.000
0129 00   7E88                          0  FD1          X,         SETCPU   ;   -SETCPU-                            8430.000
012A 00   65E8                          0  CBD          X,         TMAPR    ;   -TMAPR-                            8431.000
012B 50   4130      A          0         826          -B, AL,    RRM      ;   -TRCM-                              8432.000
012C CE   4168     19          6         82D          A.MINUS.B, AEXP, TRNM;  -TRNM-                              8433.000
012D 00   40B8                0         817          AL,        XCRM     ;   -XCRM-                              8434.000
012E 00   4F68                          0  9ED          X,         TRSC     ;   -TRSC-                              8435.000
012F 00   4F78                          0  9EF          X,         TSCR     ;   -TSCR-                              8436.000
                                    ***                                                                             8437.000
                                    *** D-ROM, PAGE 15 SECONDARY DECODE FOR ***                                      8438.000
                                    *** HARDWARE FLOATING POINT INSTRUCTIONS ***                                    8439.000
                                    *** OP CODES = E0 & E4 , Q = 38 & 39 ***                                        8440.000
                                    *** (DECODE IO BITS 12 & 3-5) ***                                               8441.000
                                    *** D-ROM.ADDR=150+IO(12,03,04,05) ***                                          8442.000
                                    ***                                                                             8443.000
0150 08   6880                8         D10  (@150)  FFPCC,    FFPSUBS  ;   -SUFW-                              8444.000
0151 08   6C80                8         D90          FFPCC,    FFPDVDS  ;   -DVFW-                              8445.000
0152 00   4D48                          9A9                    UNDEF    ;                                       8446.000
0153 00   4D48                          9A9                    UNDEF    ;                                       8447.000
0154 00   4D48                          9A9                    UNDEF    ;                                       8448.000
0155 00   4D48                          9A9                    UNDEF    ;                                       8449.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L  3 2 / 7 5  C P U                                                    D  ROM  CONTENTS
PC   D(1)  D(2)     ALU   SC    CC   PAGE  ADDR

0156   00  4D48                                9A9                UNDEF      ;                                  8450.000
0157   00  4D48                                9A9                UNDEF      ;                                  8451.000
0158   08  6780                  8             CF0          FFPCC,  FFPADDS    ;   -ADFW-                        8452.000
0159   08  6B80                  8             D70          FFPCC,  FFPMPYS    ;   -MPFW-                        8453.000
015A   00  4D48                                9A9                UNDEF      ;                                  8454.000
015B   00  4D48                                9A9                UNDEF      ;                                  8455.000
015C   00  4D48                                9A9                UNDEF      ;                                  8456.000
015D   00  4D48                                9A9                UNDEF      ;                                  8457.000
015E   00  4D48                                9A9                UNDEF      ;                                  8458.000
015F   00  4D48                                9A9                UNDEF      ;                                  8459.000
                                                          ***                                                     8460.000
                                                          *** D-ROM, PAGE 16 THIRD LEVEL DECODE FOR ***            8461.000
                                                          *** HARDWARE FLOATING PONIT, DOUBLE WORD INSTRUCTIONS ***8462.000
                                                          *** OP CODES E0 & E4 , Q = 38 & 39 ***                   8463.000
                                                          *** (DECODES I0 12 & 3-5) ***                           8464.000
                                                          *** D-ROM.ADDR=160+I0(12,03,04,05) ***                  8465.000
                                                          ***                                                     8466.000
0160   08  6A80                  8             D50  (@160)  FFPCC,  FFPSUBD    ;   -SUFD-                        8467.000
0161   08  6E88                  8             DD7          FFPCC,  FFPDVDD    ;   -DVFD-                        8468.000
0162   00  4D48                          0     9A9          X,      UNDEF      ;                                  8469.000
0163   00  4D48                          0     9A9          X,      UNDEF      ;                                  8470.000
0164   00  4D48                          0     9A9          X,      UNDEF      ;                                  8471.000
0165   00  4D48                          0     9A9          X,      UNDEF      ;                                  8472.000
0166   00  4D48                          0     9A9          X,      UNDEF      ;                                  8473.000
0167   00  4D48                          0     9A9          X,      UNDEF      ;                                  8474.000
0168   08  6980                  8             D30          FFPCC,  FFPADDD    ;   -ADFD-                        8475.000
0169   08  6D80                  8             DB0          FFPCC,  FFPMPYD    ;   -MPFD-                        8476.000
016A   00  4D48                          0     9A9          X,      UNDEF      ;                                  8477.000
016B   00  4D48                          0     9A9          X,      UNDEF      ;                                  8478.000
016C   00  4D48                          0     9A9          X,      UNDEF      ;                                  8479.000
016D   00  4D48                          0     9A9          X,      UNDEF      ;                                  8480.000
016E   00  4D48                          0     9A9          X,      UNDEF      ;                                  8481.000
016F   00  4D48                          0     9A9          X,      UNDEF      ;                                  8482.000
                                                          ***                                                     8483.000
                                                          *** D-ROM, PAGE 17 SECONDARY DECODE FOR ***              8484.000
                                                          *** REGISTER TO REGISTER TRANSFER INSTRUCTIONS ***       8485.000
                                                          *** OP CODES = 04,08,0C,10,14,38,3C ; Q = 01-05,0E,0F ***8486.000
                                                          *** (DECODE I0 12 & 3-5) ***                            8487.000
                                                          *** D-ROM.ADDR=170+I0(12,03,04,05) ***                  8488.000
                                                          ***                                                     8489.000
0170   00  4D48                          0     9A9  (@170)  X,      UNDEF      ;                                  8490.000
0171   08  4110        1          0           822          A.AND.B, AL,  RR1  ;   -ANR-                         8491.000
0172   20  4110        4          0           822          A.OR.B,  AL,  RR1  ;   -ORR-                         8492.000
0173   30  4110        6          0           822          A.EOR.B, AL,  RR1  ;   -EOR-                         8493.000
0174   CF  4140       19          7           828          A.MINUS.B, C, CAR  ;   -CAR-                         8494.000
0175   32  4120        6          2           824          A.EOR.B, E,   CMR  ;   -CMR-                         8495.000
0176   B0  4110       16          0           822          A.PLUS.B, AL, RR1  ;   -ADR-                         8496.000
0177   C8  4110       19          0           822          A.MINUS.B,AL, RR1  ;   -SUR-                         8497.000
0178   00  4D48                          0     9A9          X,      UNDEF      ;                                  8498.000
0179   00  4D48                          0     9A9          X,      UNDEF      ;                                  8499.000
017A   20  4130        4          0           826          A.OR.B, AL,   RRM  ;   -ORRM-                        8500.000
017B   30  4130        6          0           826          A.EOR.B AL,   RRM  ;   -EORM-                        8501.000
017C   00  4D48                    0     9A9                X,      UNDEF      ;                                  8502.000
017D   00  4D48                    0     9A9                X,      UNDEF      ;                                  8503.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1) D(2)     ALU  SC    CC   PAGE  ADDR
017E B6   40F0     16         6          81E          A.PLUS.B,  AEXP, RRM1;   -ADRM-                                  8504.000
017F CE   40F0     19         6          81E          A.MINUS.B, AEXP, RRM1;   -SURM-                                  8505.000
                                                  ***                                                                    8506.000
                                                  *** D-ROM, PAGE 18 SECONDARY DECODE FOR ***                             8507.000
                                                  *** BRANCH UNCONDITIONAL AND BRANCH CONDITION TRUE INSTRUCTIONS ***     8508.000
                                                  *** BRANCH FUNCTION TRUE AND BRANCH CONDITION FALSE INSTRUCTIONS ***    8509.000
                                                  *** OP CODE = EC AND F0 , Q = 38 AND 3C ***                            8510.000
                                                  *** (DECODE IO 5-8) ***                                                8511.000
                                                  *** D-ROM.ADDR=180+IO(05,06,07,08) ***                                 8512.000
                                                  ***                                                                    8513.000
0180 05   4AE0          5          95C     (@180)    S,        BFT        ;   -BFT-                                      8514.000
0181 05   4990          5          932             S,        BR1        ;   -BCF-                                      8515.000
0182 05   4990          5          932             S,        BR1        ;   -BCF-                                      8516.000
0183 05   4990          5          932             S,        BR1        ;   -BCF-                                      8517.000
0184 05   4990          5          932             S,        BR1        ;   -BCF-                                      8518.000
0185 05   4990          5          932             S,        BR1        ;   -BCF-                                      8519.000
0186 05   4990          5          932             S,        BR1        ;   -BCF-                                      8520.000
0187 05   4990          5          932             S,        BR1        ;   -BCF-                                      8521.000
0188 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8522.000
0189 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8523.000
018A 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8524.000
018B 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8525.000
018C 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8526.000
018D 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8527.000
018E 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8528.000
018F 05   4D48          5          9A9             S,        UNDEF      ;   -BU,BCT-                                   8529.000
                                                  ***                                                                    8530.000
                                                  *** D-ROM, PAGE 19 SECONDARY DECODE FOR ZMW, ZMH, ***                   8531.000
                                                  *** ZMD, ZMB, BL, BRI, LPSW, JWCS, TRP, AND TPR INSTRUCTIONS ***       8532.000
                                                  *** OP CODE = F8 , Q = 3F ***                                          8533.000
                                                  *** (DECODE IO 5-8) ***                                                8534.000
                                                  *** D-ROM.ADDR=190+IO(05,06,07,08) ***                                 8535.000
                                                  ***                                                                    8536.000
                                                  *** D-ROM, PAGE 19 THIRD LEVEL DECODE FOR ***                          8537.000
                                                  *** LOAD, STORE, CONVERT ADDRESS THRU EXTERNAL MAP (LEM, SEM, CEMA) *** 8538.000
                                                  *** OPCODE = C8 , Q = 32 ***                                           8539.000
                                                  *** (DECODE IO 12-15) ***                                              8540.000
                                                  *** D-ROM.ADDR=190+IO(12,13,14,15) ***                                 8541.000
                                                  ***                                                                    8542.000
0190 00   4648            0        8C9     (@190)    X,        ZM         ;   -ZMW,ZMH,ZMD,ZMB-                          8543.000
0191 05   49E8          5          93D             S,        BL         ;   -BL-                                       8544.000
0192 00   20E0                     41C                       BRI        ;   -BRI-                                      8545.000
0193 6F   4D80                6F   980             NP=@6F,   LPSD       ;   -LPSD-                                     8546.000
0194 69   51F8                69   A3F             NP=@69,   JWCS       ;   -JWCS- (NP=@69 IS FOR SOFTWARE             8547.000
                                                  *                                    WCS DIAGNOSTIC *JUMPD TEST)       8548.000
0195 6F   4D80                6F   980             NP=@6F,   LPSDCM     ;   -LPSDCM-                                   8549.000
0196 00   48C0                     918                       TRP        ;   -TRP-                                      8550.000
0197 00   48F8                     91F                       TPR        ;   -TPR-                                      8551.000
0198 00   4D48                     9A9                       UNDEF      ;   -SEM-                                      8552.000
0199 00   4D48                     9A9                       UNDEF      ;   -LFM-                                      8553.000
019A 00   4D48                     9A9                       UNDEF      ;   -CEMA-                                     8554.000
019B 00   4D48                     9A9                       UNDEF      ;                                              8555.000
019C 00   4D48                     9A9                       UNDEF      ;                                              8556.000
019D 00   4D48                     9A9                       UNDEF      ;                                              8557.000
```

```
                S Y S T E M S   M I C R O C O D E   A S S E M B L E R  -  R E V I S I O N  3.0  -  7 9 A P R 2 7
S E L   3 2 / 7 5   C P U                                                D   R O M   C O N T E N T S
PC   D(1)  D(2)     ALU   SC    CC    PAGE  ADDR

019E   00  4D48                                 9A9                    UNDFF      ;                                          8558.000
019F   00  4D48                                 9A9                    UNDFF      ;                                          8559.000
                                                                  ***                                                          8560.000
                                                                  *** D-ROM, PAGE 1A SECONDARY DECODE FOR INTERRUPT ***         8561.000
                                                                  *** CONTROL AND I/O INSTRUCTIONS ***                          8562.000
                                                                  *** OP CODE = FC , Q = 3F ***                                 8563.000
                                                                  *** (DECODE IO 13-15 & 18) ***                               8564.000
                                                                  *** D-ROM.ADDR=1A0+IO(13,14,15,18) ***                       8565.000
                                                                  ***                                                          8566.000
01A0   00  2208                                 441    (@1A0)          EI         ;   -EI-                                   8567.000
01A1   00  2208                                 441                    EI         ;   -EI-                                   8568.000
01A2   00  2230                                 446                    DI.INST    ;   -DI-                                   8569.000
01A3   00  2230                                 446                    DI.INST    ;   -DI-                                   8570.000
01A4   00  2250                                 44A                    RI         ;   -RI-                                   8571.000
01A5   00  2250                                 44A                    RI         ;   -RI-                                   8572.000
01A6   00  2278                                 44F                    AI         ;   -AI-                                   8573.000
01A7   00  2278                                 44F                    AI         ;   -AI-                                   8574.000
01A8   00  2280                                 456                    DAI        ;   -DAI-                                  8575.000
01A9   00  2280                                 456                    DAI        ;   -DAI-                                  8576.000
01AA   80  2498          8                      493            SRL,    TD         ;   -TD-                                   8577.000
01AB   80  2498          8                      493            SRL,    TD         ;   -TD-                                   8578.000
01AC   00  1CA8                                 395                    CD         ;   -CD-                                   8579.000
01AD   00  1CA8                                 395                    CD         ;   -CD-                                   8580.000
01AE   E8  7000                        E8       E00          NP=@E8,   EXT.IO.INST ;   -SIO,TIO,STPIO,RSCHNL,HIO,GRIO,RSCTL- 8581.000
                                                                  *                        -ECWCS,WCWCS,ECI,DCI,ACI,DACI-       8582.000
01AF   E8  7000                        E8       E00          NP=@E8,   EXT.IO.INST ;   -SIO,TIO,STPIO,RSCHNL,HIO,GRIO,RSCTL- 8583.000
                                                                  *                        -ECWCS,WCWCS,ECI,DCI,ACI,DACI-       8584.000
                                                                  ***                                                          8585.000
                                                                  *** D-ROM, PAGE 1B SECONDARY DECODE FOR ***                   8586.000
                                                                  *** SHIFT INSTRUCTIONS ***                                   8587.000
                                                                  *** OP CODES = 6C,70,74,78, & 7C ; Q = 1B-1F ***             8588.000
                                                                  *** (DECODE IO 3-5 & 9) ***                                  8589.000
                                                                  *** D-ROM.ADDR=1B0+IO(03,04,05,09) ***                       8590.000
                                                                  ***                                                          8591.000
01B0   00  4D48                        0        9A9    (@1B0)    X,         UNDEF      ;                                    8592.000
01B1   00  4D48                        0        9A9              X,         UNDEF      ;                                    8593.000
01B2   00  4D48                        0        9A9              X,         UNDEF      ;                                    8594.000
01B3   90  4260          9                      84C              SRA,       SHIFTX1    ;   -SRA-                             8595.000
01B4   80  4260          8                      84C              SRL,       SHIFTX1    ;   -SRL-                             8596.000
01B5   C0  4260          C                      84C              SRC,       SHIFTX1    ;   -SRC-                             8597.000
01B6   90  4260          9                      84C              SRAD,      SHIFTX1    ;   -SRAD-                            8598.000
01B7   80  4260          8                      84C              SRLD,      SHIFTX1    ;   -SRLD-                            8599.000
01B8   00  4D48                        0        9A9              X,         UNDEF      ;                                    8600.000
01B9   00  4D48                        0        9A9              X,         UNDEF      ;                                    8601.000
01BA   00  4D48                        0        9A9              X,         UNDEF      ;                                    8602.000
01BB   70  4260          7                      850              SLA,       SHIFTX2    ;   -SLA-                             8603.000
01BC   00  4260          0                      84C              SLL,       SHIFTX1    ;   -SLL-                             8604.000
01BD   10  4260          1                      84C              SLC,       SHIFTX1    ;   -SLC-                             8605.000
01BE   60  4280          6                      850              SLAD,      SHIFTX2    ;   -SLAD-                            8606.000
01BF   20  4260          2                      84C              SLLD,      SHIFTX1    ;   -SLLD-                            8607.000
                                                                  ***                                                          8608.000
                                                                  *** D-ROM, PAGE 1C SECONDARY DECODE FOR ***                   8609.000
                                                                  *** BIB, BIH, BIW, AND BID INSTRUCTIONS ***                   8610.000
                                                                  *** OP CODE = F4 , Q = 3D ***                                8611.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU D ROM CONTENTS

```
PC   D(1)  D(2)     ALU   SC    CC   PAGE  ADDR

                                                *** (DECODF I0 9 & 10) ***                                          8612.000
                                                *** D-ROM.ADDR=1C0+I0(09,10)*4 ***                                  8613.000
                                                ***                                                                 8614.000
01C0   00  4A30                          946    (@1C0)          BIB       ;   -BIB-                                 8615.000
01C1   00  4D48                          9A9                    UNDEF     ;                                         8616.000
01C2   00  4D48                          9A9                    UNDEF     ;                                         8617.000
01C3   00  4D48                          9A9                    UNDEF     ;                                         8618.000
01C4   00  4A38                          947                    BIH       ;   -BIH-                                 8619.000
01C5   00  4D48                          9A9                    UNDEF     ;                                         8620.000
01C6   00  4D48                          9A9                    UNDEF     ;                                         8621.000
01C7   00  4D48                          9A9                    UNDEF     ;                                         8622.000
01C8   00  4A40                          948                    BIW       ;   -BIW-                                 8623.000
01C9   00  4D48                          9A9                    UNDEF     ;                                         8624.000
01CA   00  4D48                          9A9                    UNDEF     ;                                         8625.000
01CB   00  4D48                          9A9                    UNDEF     ;                                         8626.000
01CC   00  4A48                          949                    BID       ;   -BID-                                 8627.000
01CD   00  4D48                          9A9                    UNDEF     ;                                         8628.000
01CE   00  4D48                          9A9                    UNDEF     ;                                         8629.000
01CF   00  4D48                          9A9                    UNDEF     ;                                         8630.000
                                                ***                                                                 8631.000
                                                *** D-ROM, PAGE 1E SECONDARY DECODE FOR ***                          8632.000
                                                *** HARDWARE AND FIRMWARE FLOATING POINT INSTRUCTIONS ***            8633.000
                                                *** OP CODE = E0 , Q = 38 ***                                       8634.000
                                                *** (DECODE I0 BITS 12 & 30 AND HARDWARE FP BIT) ***                8635.000
                                                *** D-ROM.ADDR=1E0+I0(12,30)*4+%HARDFP*2 ***                        8636.000
                                                ***                                                                 8637.000
01E0   08  6880                    8     D10    (@1E0)  FFPCC,    FFPSUBS   ;   -SUFW-                           8638.000
01E1   00  4D48                          9A9                    UNDEF     ;                                         8639.000
01E2   6C  3008                    6C    601            NP=@6C,  FIRMWARE.FP ;   -SUFW-                           8640.000
01E3   00  4D48                          9A9                    UNDEF     ;                                         8641.000
01E4   08  6A80                    8     D50            FFPCC,   FFPSUBD   ;   -SUFD-                             8642.000
01E5   00  4D48                          9A9                    UNDEF     ;                                         8643.000
01E6   6C  3008                    6C    601            NP=@6C,  FIRMWARE.FP ;   -SUFD-                           8644.000
01E7   00  4D48                          9A9                    UNDEF     ;                                         8645.000
01E8   08  6780                    8     CF0            FFPCC,   FFPADDS   ;   -ADFW-                             8646.000
01E9   00  4D48                          9A9                    UNDEF     ;                                         8647.000
01EA   6C  3008                    6C    601            NP=@6C,  FIRMWARE.FP ;   -ADFW-                           8648.000
01EB   00  4D48                          9A9                    UNDEF     ;                                         8649.000
01EC   08  6980                    8     D30            FFPCC,   FFPADDD   ;   -ADFD-                             8650.000
01ED   00  4D48                          9A9                    UNDEF     ;                                         8651.000
01EE   6C  3008                    6C    601            NP=@6C   FIRMWARE.FP ;   -ADFD-                           8652.000
01EF   00  4D48                          9A9                    UNDEF     ;                                         8653.000
                                                ***                                                                 8654.000
                                                *** D-ROM, PAGE 1F SECONDARY DECODE FOR ***                          8655.000
                                                *** HARDWARE AND FIRMWARE FLOATING POINT INSTRUCTIONS ***            8656.000
                                                *** OP CODE = E4 , Q = 39 ***                                       8657.000
                                                *** (DECODE I0 BITS 12 & 30 AND HARDWARE FP BIT) ***                8658.000
                                                *** D-ROM.ADDR=1F0+I0(12,30)*4+%HARDFP*2 ***                        8659.000
                                                ***                                                                 8660.000
01F0   08  6C80                    8     D90    (@1F0)  FFPCC,    FFPDVDS   ;   -DVFW-                           8661.000
01F1   00  4D48                          9A9                    UNDEF     ;                                         8662.000
01F2   6C  3008                    6C    601            NP=@6C,  FIRMWARE.FP ;   -DVFW-                           8663.000
01F3   00  4D48                          9A9                    UNDEF     ;                                         8664.000
01F4   08  6E88                    8     DD7            FFPCC,   FFPDVDD   ;   -DVFD-                             8665.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU DROM CONTENTS

PC	D(1)	D(2)	ALU	SC	CC	PAGE	ADDR				
01F5	00	4D48					9A9		UNDEF	;	8666.000
01F6	6C	3008				6C	601	NP=@6C,	FIRMWARE.FP	; -DVFD-	8667.000
01F7	00	4D48					9A9		UNDEF	;	8668.000
01F8	0A	6B80			8		D70	FFPCC,	FFPMPYS	; -MPFW-	8669.000
01F9	00	4D48					9A9		UNDEF	;	8670.000
01FA	6C	3008				6C	601	NP=@6C,	FIRMWARE.FP	; -MPFW-	8671.000
01FB	00	4D48					9A9		UNDEF	;	8672.000
01FC	0A	6D80			8		D80	FFPCC,	FFPMPYD	; -MPFD-	8673.000
01FD	00	4D48					9A9		UNDEF	;	8674.000
01FE	6C	3008				6C	601	NP=@6C	FIRMWARE.FP	; -MPFD-	8675.000
01FF	00	4D48					9A9		UNDEF	;	8676.000
									$END		8677.000

E R R O R M E S S A G E S & S T A T I S T I C S

```
SYMBOL TABLE OCCUPIED  33160 BYTES (08188 HEX)
WORKSPACE STACK USED   67128 BYTES (10638 HEX)
UNUSED CORE SPACE WAS  35648 BYTES (08B40 HEX)
MAXIMUM PATTERN FAILS WAS    229
NUMBER OF CROSS REFERENCE ITEMS WAS  23308
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

```
!                       STRING   REFERENCED AT :    858.000  884.000 1650.000 1768.000 1783.000 2291.000 2460.000 2727.000
                                 2732.000 2741.000 2742.000 2745.000 2746.000 3762.000 3822.000 3849.000 3898.000 3912.000
                                 4103.000 4184.000 4746.000 4752.000 4833.000 5798.000 5801.000 5982.000 5994.000 6008.000
                                 6066.000 6090.000 6116.000 6160.000 6193.000 6206.000 6267.000 6290.000 6670.000 6676.000
                                 7232.000 7537.000 7602.000 7683.000 7701.000 7726.000

#                       STRING   REFERENCED AT :    639.000  683.000 1092.000 1097.000 2890.000 2900.000 3633.000 4077.000
                                 4076.000 4501.000 4507.000 4514.000 4516.000 4518.000 4520.000 4526.000 4541.000 4545.000
                                 4546.000 4553.000 4562.000 4570.000 4574.000 4657.000 4661.000 4674.000 4687.000 4816.000
                                 4825.000 4870.000 4877.000 4895.000 4975.000 5099.000 5142.000 5176.000 5178.000 5199.000
                                 5202.000 5224.000 5296.000 5330.000 6719.000 7955.000

$                       VARIABLE REFERENCED AT :     51.000   94.000  210.000  671.000  673.000  675.000  712.000  714.000
                                  716.000  718.000  739.000  741.000  743.000  746.000  749.000  765.000  769.000  772.000
                                  774.000  801.000  814.000  840.000  866.000  871.000  944.000  948.000  968.000  979.000
                                  997.000 1009.000 1013.000 1015.000 1023.000 1036.000 1134.000 1135.000 1136.000 1272.000
                                 1276.000 1279.000 1281.000 1322.000 1477.000 1512.000 1518.000 1573.000 1608.000 1611.000
                                 1616.000 1627.000 1681.000 1690.000 1702.000 1768.000 1777.000 1794.000 1857.000 1891.000
                                 1892.000 1950.000 1951.000 1975.000 2045.000 2046.000 2049.000 2050.000 2052.000 2053.000
                                 2079.000 2126.000 2128.000 2132.000 2137.000 2140.000 2141.000 2148.000 2151.000 2189.000
                                 2205.000 2211.000 2216.000 2224.000 2233.000 2277.000 2296.000 2338.000 2429.000 2448.000
                                 2449.000 2457.000 2458.000 2491.000 2516.000 2519.000 2522.000 2544.000 2635.000 2672.000
                                 2765.000 2770.000 2773.000 2781.000 2807.000 2836.000 2928.000 2979.000 2980.000 3062.000
                                 3165.000 3218.000 3227.000 3267.000 3316.000 3385.000 3419.000 3425.000 3521.000 3537.000
                                 3552.000 3676.000 3730.000 3744.000 3757.000 3774.000 3781.000 3783.000 3829.000 3833.000
                                 3837.000 3899.000 3904.000 3926.000 3963.000 4030.000 4038.000 4062.000 4093.000 4097.000
                                 4107.000 4111.000 4133.000 4138.000 4145.000 4219.000 4248.000 4264.000 4268.000 4273.000
                                 4282.000 4287.000 4300.000 4304.000 4435.000 4436.000 4457.000 4469.000 4473.000 4558.000
                                 4562.000 4566.000 4568.000 4570.000 4577.000 4610.000 4636.000 4764.000 4785.000 4848.000
                                 4879.000 4896.000 4918.000 4943.000 4948.000 4950.000 5023.000 5068.000 5077.000 5090.000
                                 5095.000 5126.000 5127.000 5128.000 5132.000 5137.000 5160.000 5170.000 5243.000 5309.000
                                 5432.000 5466.000 5504.000 5535.000 5541.000 5544.000 5738.000 5749.000 5843.000 5898.000
                                 5903.000 5904.000 5905.000 5906.000 5912.000 5934.000 5946.000 5947.000 5996.000 5997.000
                                 6053.000 6077.000 6097.000 6102.000 6108.000 6153.000 6155.000 6276.000 6278.000 6280.000
                                 6282.000 6328.000 6340.000 6342.000 6344.000 6346.000 6348.000 6364.000 6366.000 6396.000
                                 6408.000 6497.000 6587.000 6658.000 6664.000 6694.000 6805.000 6811.000 6868.000 6873.000
                                 6930.000 6993.000 7001.000 7049.000 7065.000 7104.000 7139.000 7157.000 7201.000 7203.000
                                 7205.000 7234.000 7277.000 7286.000 7440.000 7456.000 7486.000 7494.000 7500.000 7507.000
                                 7517.000 7524.000 7531.000 7558.000 7567.000 7573.000 7577.000 7580.000 7592.000 7599.000
                                 7655.000 7666.000 7679.000 7702.000 7716.000 7818.000 7825.000 7846.000 7866.000 7896.000
                                 7898.000 7927.000 7950.000 7985.000 7989.000 7991.000 8020.000 8023.000 8034.000 8059.000
                                 8069.000

%                       STRING   REFERENCED AT :     51.000   94.000  210.000  603.000  651.000  673.000  675.000  677.000
                                  693.000  698.000  705.000  712.000  714.000  716.000  718.000  739.000  741.000  743.000
                                  746.000  749.000  765.000  769.000  772.000  774.000  787.000  812.000  814.000  821.000
                                  822.000  826.000  828.000  835.000  840.000  846.000  847.000  849.000  872.000  907.000
                                  931.000  944.000  964.000  966.000  968.000  997.000 1005.000 1009.000 1012.000 1013.000
                                 1023.000 1024.000 1044.000 1048.000 1050.000 1062.000 1063.000 1072.000 1074.000 1086.000
                                 1087.000 1088.000 1089.000 1089.000 1103.000 1113.000 1120.000 1128.000 1135.000 1151.000
                                 1165.000 1186.000 1259.000 1268.000 1272.000 1276.000 1312.000 1316.000 1322.000 1327.000
                                 1349.000 1444.000 1477.000 1512.000 1518.000 1522.000 1576.000 1582.000 1593.000 1616.000
                                 1627.000 1645.000 1664.000 1680.000 1685.000 1702.000 1740.000 1752.000 1755.000 1756.000
                                 1757.000 1760.000 1768.000 1802.000 1809.000 1891.000 1900.000 1902.000 1910.000 1939.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

1951.000 2050.000 2066.000 2069.000 2079.000 2092.000 2126.000 2127.000 2128.000 2132.000
2137.000 2141.000 2142.000 2148.000 2151.000 2152.000 2168.000 2179.000 2181.000 2193.000
2201.000 2205.000 2206.000 2207.000 2211.000 2216.000 2217.000 2218.000 2234.000 2235.000
2244.000 2248.000 2256.000 2261.000 2271.000 2272.000 2277.000 2294.000 2301.000 2364.000
2411.000 2422.000 2429.000 2436.000 2454.000 2472.000 2491.000 2516.000 2522.000 2540.000
2544.000 2548.000 2563.000 2587.000 2621.000 2668.000 2681.000 2688.000 2714.000 2719.000
2720.000 2734.000 2755.000 2756.000 2765.000 2781.000 2789.000 2815.000 2816.000 2850.000
2851.000 2880.000 2881.000 2905.000 2928.000 2946.000 2965.000 2971.000 2980.000 2999.000
3002.000 3005.000 3026.000 3053.000 3056.000 3059.000 3067.000 3122.000 3124.000 3146.000
3165.000 3172.000 3227.000 3267.000 3269.000 3357.000 3358.000 3382.000 3385.000 3414.000
3419.000 3450.000 3461.000 3472.000 3487.000 3497.000 3516.000 3521.000 3523.000 3535.000
3537.000 3547.000 3554.000 3557.000 3582.000 3600.000 3658.000 3717.000 3737.000 3751.000
3771.000 3774.000 3778.000 3783.000 3833.000 3837.000 3878.000 3879.000 3904.000 3963.000
3994.000 3998.000 4001.000 4030.000 4062.000 4080.000 4138.000 4216.000 4228.000 4242.000
4257.000 4267.000 4269.000 4270.000 4376.000 4409.000 4410.000 4412.000 4424.000 4457.000
4464.000 4469.000 4473.000 4473.000 4480.000 4487.000 4500.000 4511.000 4525.000 4534.000
4534.000 4539.000 4540.000 4544.000 4545.000 4550.000 4558.000 4562.000 4566.000 4570.000
4573.000 4577.000 4584.000 4588.000 4596.000 4597.000 4632.000 4636.000 4638.000 4639.000
4649.000 4659.000 4682.000 4710.000 4723.000 4729.000 4741.000 4750.000 4785.000 4817.000
4826.000 4834.000 4844.000 4848.000 4850.000 4851.000 4871.000 4879.000 4881.000 4881.000
4882.000 4892.000 4896.000 4918.000 4936.000 4943.000 4964.000 4964.000 4970.000 4970.000
4971.000 4977.000 4993.000 4993.000 4999.000 4999.000 5000.000 5013.000 5023.000 5039.000
5061.000 5068.000 5077.000 5078.000 5089.000 5090.000 5092.000 5112.000 5130.000 5132.000
5160.000 5170.000 5186.000 5188.000 5189.000 5192.000 5193.000 5212.000 5216.000 5223.000
5243.000 5243.000 5255.000 5280.000 5297.000 5303.000 5309.000 5313.000 5343.000 5345.000
5347.000 5354.000 5355.000 5416.000 5437.000 5439.000 5452.000 5466.000 5504.000 5510.000
5535.000 5541.000 5544.000 5568.000 5571.000 5618.000 5738.000 5749.000 5760.000 5812.000
5814.000 5815.000 5816.000 5817.000 5886.000 5899.000 5903.000 5905.000 5912.000 5926.000
5947.000 6039.000 6045.000 6121.000 6128.000 6150.000 6210.000 6242.000 6243.000 6244.000
6245.000 6274.000 6317.000 6339.000 6383.000 6385.000 6394.000 6396.000 6399.000 6401.000
6405.000 6408.000 6412.000 6417.000 6418.000 6420.000 6425.000 6465.000 6471.000 6480.000
6497.000 6541.000 6543.000 6567.000 6587.000 6595.000 6643.000 6658.000 6694.000 6695.000
6719.000 6732.000 6745.000 6755.000 6805.000 6868.000 6880.000 6937.000 7001.000 7065.000
7104.000 7139.000 7157.000 7203.000 7207.000 7327.000 7335.000 7341.000 7344.000 7348.000
7440.000 7453.000 7461.000 7477.000 7481.000 7487.000 7492.000 7500.000 7515.000 7524.000
7529.000 7533.000 7549.000 7553.000 7559.000 7573.000 7577.000 7590.000 7599.000 7604.000
7655.000 7662.000 7664.000 7696.000 7702.000 7716.000 7731.000 7740.000 7809.000 7810.000
7828.000 7950.000 7955.000 7958.000 7989.000 8052.000 8054.000 8055.000 8059.000

%BLKTIMEOUT STRING REFERENCED AT : 722.000

%WCS STRING REFERENCED AT : 5543.000 5566.000 5592.000

R STRING REFERENCED AT : 694.000 695.000 698.000 699.000 705.000 786.000 787.000 820.000
864.000 869.000 908.000 909.000 911.000 919.000 936.000 948.000 952.000 953.000
955.000 998.000 1002.000 1005.000 1027.000 1045.000 1050.000 1056.000 1062.000 1072.000
1074.000 1086.000 1088.000 1089.000 1125.000 1255.000 1345.000 1361.000 1363.000 1365.000
1450.000 1453.000 1532.000 1549.000 1571.000 1574.000 1576.000 1607.000 1608.000 1648.000
1675.000 1678.000 1681.000 1686.000 1723.000 1739.000 1752.000 1755.000 1767.000 1772.000
1781.000 1807.000 1815.000 1816.000 1817.000 1838.000 1839.000 1901.000 1904.000 1951.000
2041.000 2076.000 2086.000 2117.000 2143.000 2166.000 2167.000 2222.000 2223.000 2228.000
2237.000 2262.000 2264.000 2268.000 2290.000 2315.000 2324.000 2327.000 2343.000 2498.000
2520.000 2537.000 2538.000 2552.000 2561.000 2587.000 2608.000 2620.000 2621.000 2664.000
2666.000 2667.000 2668.000 2672.000 2688.000 2767.000 2768.000 2772.000 2805.000 2825.000
2834.000 2838.000 2860.000 2871.000 2888.000 2905.000 2930.000 2942.000 2956.000 2958.000

MICROCODE CROSS-REFERENCE LIST

SFL 32/75 CPU

2967.000 3124.000 3146.000 3159.000 3160.000 3164.000 3166.000 3171.000 3216.000 3220.000
3224.000 3243.000 3306.000 3388.000 3406.000 3424.000 3443.000 3452.000 3464.000 3479.000
3530.000 3542.000 3560.000 3563.000 3743.000 3795.000 3807.000 3812.000 3911.000 3992.000
3994.000 3998.000 4000.000 4035.000 4141.000 4217.000 4245.000 4262.000 4278.000 4284.000
4285.000 4404.000 4476.000 4498.000 4499.000 4508.000 4517.000 4519.000 4531.000 4539.000
4544.000 4558.000 4566.000 4577.000 4616.000 4708.000 4712.000 4715.000 4744.000 4748.000
4776.000 4784.000 4790.000 4908.000 4914.000 4923.000 4977.000 5004.000 5111.000 5112.000
5114.000 5157.000 5242.000 5249.000 5250.000 5308.000 5314.000 5324.000 5351.000 5360.000
5363.000 5366.000 5416.000 5439.000 5462.000 5544.000 5545.000 5546.000 5566.000 5571.000
5726.000 5731.000 5746.000 5749.000 5793.000 5803.000 5832.000 5838.000 5866.000 5876.000
5932.000 5934.000 5940.000 5954.000 5955.000 5985.000 5986.000 6014.000 6015.000 6026.000
6029.000 6049.000 6062.000 6070.000 6071.000 6074.000 6078.000 6091.000 6092.000 6095.000
6096.000 6106.000 6139.000 6148.000 6167.000 6170.000 6179.000 6183.000 6184.000 6199.000
6225.000 6228.000 6236.000 6258.000 6266.000 6333.000 6354.000 6359.000 6389.000 6442.000
6465.000 6468.000 6480.000 6481.000 6533.000 6666.000 6668.000 6669.000 6671.000 6675.000
6741.000 6868.000 7201.000 7207.000 7209.000 7233.000 7243.000 7255.000 7256.000 7321.000
7327.000 7335.000 7423.000 7536.000 7617.000 7637.000 7651.000 7682.000 7693.000 7725.000
7746.000 7750.000 7833.000 7882.000 7982.000 7983.000 7987.000 8032.000 8033.000 8035.000
8036.000 8048.000

&% STRING REFERENCED AT : 905.000 1386.000 1894.000 2678.000 4266.000 6158.000 6159.000 6252.000
6535.000

* STRING REFERENCED AT : 51.000 94.000 210.000 594.000 603.000 639.000 651.000 660.000
671.000 672.000 673.000 674.000 675.000 676.000 677.000 678.000 683.000 685.000
693.000 696.000 697.000 698.000 700.000 701.000 702.000 705.000 712.000 713.000
714.000 715.000 716.000 717.000 718.000 720.000 722.000 723.000 739.000 740.000
741.000 742.000 743.000 745.000 746.000 747.000 748.000 749.000 750.000 751.000
765.000 768.000 769.000 771.000 772.000 773.000 774.000 776.000 777.000 788.000
789.000 801.000 802.000 803.000 814.000 815.000 817.000 818.000 821.000 822.000
824.000 825.000 826.000 827.000 828.000 833.000 835.000 839.000 840.000 842.000
845.000 846.000 847.000 848.000 849.000 854.000 856.000 860.000 866.000 871.000
872.000 874.000 884.000 897.000 907.000 910.000 921.000 922.000 924.000 926.000
930.000 931.000 933.000 935.000 938.000 944.000 946.000 948.000 957.000 964.000
968.000 969.000 970.000 971.000 979.000 980.000 986.000 996.000 997.000 997.000
1000.000 1004.000 1006.000 1009.000 1010.000 1011.000 1012.000 1013.000 1014.000 1015.000
1016.000 1017.000 1018.000 1020.000 1021.000 1022.000 1023.000 1024.000 1036.000 1037.000
1038.000 1040.000 1041.000 1044.000 1047.000 1048.000 1052.000 1059.000 1060.000 1064.000
1072.000 1073.000 1074.000 1075.000 1076.000 1078.000 1089.000 1091.000 1092.000 1095.000
1097.000 1105.000 1109.000 1113.000 1114.000 1116.000 1120.000 1121.000 1123.000 1129.000
1134.000 1135.000 1136.000 1138.000 1156.000 1186.000 1197.000 1206.000 1209.000 1218.000
1220.000 1222.000 1226.000 1229.000 1235.000 1237.000 1238.000 1255.000 1259.000 1261.000
1265.000 1268.000 1272.000 1273.000 1275.000 1276.000 1279.000 1281.000 1285.000 1295.000
1303.000 1306.000 1307.000 1309.000 1311.000 1312.000 1314.000 1315.000 1316.000 1320.000
1322.000 1323.000 1324.000 1325.000 1327.000 1328.000 1329.000 1330.000 1334.000 1336.000
1339.000 1341.000 1343.000 1346.000 1349.000 1354.000 1357.000 1359.000 1364.000 1367.000
1368.000 1370.000 1372.000 1375.000 1377.000 1378.000 1387.000 1392.000 1395.000 1399.000
1401.000 1402.000 1408.000 1430.000 1439.000 1440.000 1441.000 1444.000 1445.000 1448.000
1449.000 1451.000 1458.000 1459.000 1461.000 1462.000 1466.000 1468.000 1470.000 1473.000
1475.000 1477.000 1481.000 1482.000 1484.000 1492.000 1499.000 1500.000 1512.000 1513.000
1514.000 1516.000 1518.000 1519.000 1523.000 1524.000 1537.000 1552.000 1553.000 1554.000
1557.000 1573.000 1575.000 1576.000 1578.000 1580.000 1582.000 1583.000 1585.000 1593.000
1594.000 1608.000 1611.000 1613.000 1616.000 1618.000 1620.000 1622.000 1627.000 1630.000
1644.000 1645.000 1646.000 1651.000 1652.000 1653.000 1656.000 1657.000 1664.000 1665.000
1678.000 1681.000 1685.000 1690.000 1691.000 1692.000 1701.000 1702.000 1703.000 1706.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

1710.000 1712.000 1716.000 1725.000 1727.000 1730.000 1732.000 1734.000 1735.000 1739.000
1740.000 1742.000 1743.000 1747.000 1749.000 1754.000 1756.000 1757.000 1758.000 1759.000
1760.000 1761.000 1763.000 1765.000 1768.000 1769.000 1770.000 1777.000 1782.000 1785.000
1786.000 1788.000 1790.000 1794.000 1795.000 1797.000 1800.000 1801.000 1802.000 1803.000
1805.000 1808.000 1809.000 1813.000 1817.000 1818.000 1819.000 1821.000 1823.000 1842.000
1844.000 1850.000 1851.000 1857.000 1858.000 1864.000 1867.000 1891.000 1892.000 1897.000
1902.000 1906.000 1908.000 1910.000 1911.000 1933.000 1937.000 1939.000 1940.000 1950.000
1951.000 1955.000 1964.000 1975.000 1977.000 1984.000 1991.000 1998.000 2005.000 2009.000
2015.000 2023.000 2024.000 2025.000 2028.000 2029.000 2045.000 2046.000 2049.000 2050.000
2051.000 2052.000 2053.000 2058.000 2066.000 2067.000 2068.000 2069.000 2070.000 2077.000
2078.000 2079.000 2080.000 2089.000 2090.000 2092.000 2099.000 2101.000 2121.000 2124.000
2125.000 2126.000 2127.000 2128.000 2129.000 2130.000 2132.000 2133.000 2134.000 2136.000
2137.000 2138.000 2139.000 2140.000 2141.000 2142.000 2144.000 2145.000 2147.000 2148.000
2149.000 2150.000 2151.000 2152.000 2153.000 2163.000 2164.000 2167.000 2168.000 2169.000
2179.000 2180.000 2181.000 2183.000 2185.000 2186.000 2188.000 2189.000 2191.000 2193.000
2194.000 2201.000 2202.000 2204.000 2205.000 2206.000 2207.000 2208.000 2209.000 2211.000
2212.000 2213.000 2215.000 2216.000 2217.000 2218.000 2219.000 2220.000 2224.000 2225.000
2226.000 2230.000 2231.000 2233.000 2234.000 2235.000 2239.000 2240.000 2243.000 2244.000
2245.000 2247.000 2248.000 2249.000 2256.000 2257.000 2260.000 2263.000 2267.000 2269.000
2270.000 2271.000 2272.000 2277.000 2278.000 2279.000 2293.000 2294.000 2296.000 2300.000
2301.000 2317.000 2326.000 2331.000 2338.000 2341.000 2344.000 2351.000 2364.000 2372.000
2377.000 2381.000 2382.000 2402.000 2406.000 2409.000 2410.000 2411.000 2413.000 2415.000
2422.000 2427.000 2428.000 2429.000 2430.000 2432.000 2433.000 2436.000 2437.000 2447.000
2448.000 2449.000 2451.000 2453.000 2454.000 2456.000 2457.000 2458.000 2459.000 2462.000
2464.000 2470.000 2472.000 2478.000 2490.000 2491.000 2492.000 2494.000 2496.000 2501.000
2512.000 2516.000 2518.000 2519.000 2522.000 2523.000 2526.000 2527.000 2530.000 2531.000
2533.000 2536.000 2539.000 2544.000 2545.000 2546.000 2547.000 2548.000 2552.000 2564.000
2567.000 2573.000 2574.000 2578.000 2584.000 2585.000 2587.000 2593.000 2594.000 2597.000
2599.000 2605.000 2606.000 2610.000 2612.000 2618.000 2619.000 2622.000 2624.000 2625.000
2628.000 2630.000 2633.000 2635.000 2636.000 2637.000 2640.000 2668.000 2672.000 2673.000
2679.000 2681.000 2682.000 2686.000 2688.000 2690.000 2691.000 2692.000 2693.000 2697.000
2701.000 2713.000 2714.000 2716.000 2718.000 2719.000 2720.000 2723.000 2726.000 2729.000
2730.000 2731.000 2734.000 2735.000 2738.000 2739.000 2740.000 2743.000 2744.000 2747.000
2748.000 2755.000 2756.000 2765.000 2770.000 2772.000 2773.000 2776.000 2778.000 2780.000
2781.000 2782.000 2789.000 2796.000 2797.000 2804.000 2807.000 2809.000 2815.000 2816.000
2823.000 2827.000 2828.000 2833.000 2836.000 2840.000 2842.000 2850.000 2851.000 2858.000
2862.000 2863.000 2869.000 2872.000 2873.000 2880.000 2881.000 2890.000 2897.000 2900.000
2905.000 2906.000 2915.000 2918.000 2928.000 2931.000 2949.000 2958.000 2962.000 2965.000
2971.000 2979.000 2980.000 2981.000 2999.000 3000.000 3002.000 3003.000 3005.000 3006.000
3010.000 3026.000 3027.000 3030.000 3034.000 3053.000 3054.000 3056.000 3057.000 3059.000
3060.000 3062.000 3067.000 3068.000 3082.000 3095.000 3103.000 3105.000 3108.000 3110.000
3115.000 3122.000 3127.000 3128.000 3130.000 3133.000 3135.000 3137.000 3143.000 3147.000
3153.000 3154.000 3157.000 3161.000 3162.000 3165.000 3172.000 3174.000 3176.000 3178.000
3181.000 3202.000 3204.000 3218.000 3219.000 3222.000 3227.000 3229.000 3232.000 3244.000
3246.000 3267.000 3268.000 3269.000 3274.000 3300.000 3306.000 3316.000 3327.000 3347.000
3350.000 3351.000 3353.000 3355.000 3357.000 3358.000 3359.000 3382.000 3385.000 3389.000
3390.000 3407.000 3409.000 3416.000 3419.000 3421.000 3425.000 3426.000 3427.000 3442.000
3443.000 3450.000 3451.000 3452.000 3460.000 3461.000 3463.000 3472.000 3474.000 3475.000
3476.000 3479.000 3487.000 3489.000 3497.000 3499.000 3504.000 3505.000 3518.000 3521.000
3522.000 3523.000 3525.000 3532.000 3535.000 3536.000 3537.000 3538.000 3544.000 3547.000
3548.000 3552.000 3554.000 3555.000 3557.000 3558.000 3560.000 3561.000 3568.000 3570.000
3572.000 3574.000 3576.000 3582.000 3585.000 3594.000 3595.000 3599.000 3600.000 3601.000
3622.000 3631.000 3633.000 3635.000 3637.000 3638.000 3641.000 3644.000 3645.000 3646.000
3649.000 3651.000 3658.000 3661.000 3663.000 3680.000 3683.000 3685.000 3687.000 3691.000
3696.000 3699.000 3702.000 3705.000 3706.000 3708.000 3711.000 3716.000 3717.000 3718.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

3720.000 3723.000 3729.000 3730.000 3731.000 3737.000 3738.000 3744.000 3745.000 3751.000
3752.000 3757.000 3758.000 3771.000 3774.000 3779.000 3781.000 3783.000 3786.000 3790.000
3797.000 3802.000 3809.000 3810.000 3818.000 3820.000 3822.000 3824.000 3825.000 3827.000
3828.000 3833.000 3835.000 3837.000 3838.000 3841.000 3879.000 3881.000 3890.000 3891.000
3898.000 3899.000 3901.000 3904.000 3907.000 3910.000 3913.000 3926.000 3943.000 3963.000
3966.000 3967.000 3968.000 3969.000 3970.000 3972.000 3975.000 3977.000 3980.000 3981.000
3984.000 3985.000 3988.000 3991.000 4001.000 4004.000 4012.000 4013.000 4019.000 4020.000
4029.000 4030.000 4031.000 4037.000 4038.000 4039.000 4050.000 4055.000 4062.000 4063.000
4068.000 4073.000 4076.000 4080.000 4085.000 4095.000 4097.000 4103.000 4104.000 4107.000
4109.000 4111.000 4113.000 4118.000 4120.000 4122.000 4124.000 4131.000 4133.000 4138.000
4139.000 4142.000 4143.000 4145.000 4147.000 4153.000 4185.000 4186.000 4187.000 4193.000
4198.000 4199.000 4200.000 4208.000 4210.000 4215.000 4216.000 4219.000 4222.000 4226.000
4228.000 4229.000 4231.000 4233.000 4235.000 4240.000 4242.000 4244.000 4248.000 4251.000
4255.000 4257.000 4264.000 4267.000 4268.000 4271.000 4273.000 4278.000 4280.000 4286.000
4287.000 4288.000 4289.000 4298.000 4300.000 4301.000 4302.000 4304.000 4306.000 4308.000
4365.000 4376.000 4380.000 4388.000 4390.000 4397.000 4401.000 4406.000 4409.000 4410.000
4412.000 4413.000 4422.000 4424.000 4430.000 4431.000 4435.000 4436.000 4437.000 4438.000
4443.000 4452.000 4453.000 4457.000 4458.000 4459.000 4464.000 4465.000 4469.000 4473.000
4476.000 4481.000 4482.000 4487.000 4488.000 4493.000 4500.000 4501.000 4502.000 4504.000
4508.000 4511.000 4512.000 4514.000 4515.000 4517.000 4519.000 4520.000 4525.000 4526.000
4527.000 4531.000 4534.000 4535.000 4537.000 4540.000 4541.000 4542.000 4545.000 4546.000
4547.000 4550.000 4551.000 4553.000 4555.000 4558.000 4559.000 4561.000 4562.000 4563.000
4566.000 4567.000 4568.000 4569.000 4570.000 4571.000 4573.000 4574.000 4575.000 4577.000
4579.000 4583.000 4585.000 4586.000 4587.000 4588.000 4590.000 4591.000 4594.000 4596.000
4598.000 4599.000 4600.000 4601.000 4603.000 4604.000 4606.000 4610.000 4613.000 4619.000
4630.000 4632.000 4636.000 4637.000 4638.000 4639.000 4640.000 4649.000 4655.000 4657.000
4659.000 4661.000 4670.000 4674.000 4682.000 4685.000 4687.000 4695.000 4696.000 4704.000
4708.000 4710.000 4713.000 4715.000 4723.000 4725.000 4727.000 4729.000 4731.000 4733.000
4741.000 4743.000 4748.000 4750.000 4752.000 4760.000 4762.000 4763.000 4764.000 4768.000
4779.000 4785.000 4789.000 4790.000 4796.000 4817.000 4818.000 4826.000 4827.000 4834.000
4835.000 4844.000 4848.000 4849.000 4850.000 4851.000 4852.000 4861.000 4869.000 4871.000
4873.000 4879.000 4880.000 4881.000 4882.000 4883.000 4892.000 4896.000 4897.000 4899.000
4907.000 4909.000 4918.000 4922.000 4923.000 4928.000 4936.000 4938.000 4943.000 4944.000
4946.000 4948.000 4950.000 4953.000 4962.000 4964.000 4969.000 4970.000 4971.000 4972.000
4979.000 4982.000 4991.000 4993.000 4998.000 4999.000 5000.000 5001.000 5004.000 5013.000
5014.000 5023.000 5024.000 5028.000 5040.000 5048.000 5052.000 5053.000 5061.000 5068.000
5069.000 5070.000 5077.000 5078.000 5079.000 5081.000 5089.000 5090.000 5091.000 5092.000
5093.000 5095.000 5096.000 5097.000 5099.000 5100.000 5112.000 5113.000 5115.000 5116.000
5117.000 5124.000 5125.000 5126.000 5127.000 5128.000 5130.000 5131.000 5132.000 5133.000
5135.000 5136.000 5137.000 5138.000 5140.000 5142.000 5143.000 5147.000 5154.000 5159.000
5160.000 5162.000 5163.000 5170.000 5171.000 5174.000 5175.000 5176.000 5177.000 5178.000
5185.000 5186.000 5187.000 5189.000 5190.000 5193.000 5194.000 5195.000 5197.000 5198.000
5199.000 5200.000 5202.000 5211.000 5212.000 5216.000 5217.000 5219.000 5223.000 5225.000
5233.000 5234.000 5238.000 5241.000 5243.000 5247.000 5251.000 5252.000 5253.000 5254.000
5255.000 5257.000 5258.000 5261.000 5269.000 5273.000 5274.000 5275.000 5277.000 5280.000
5281.000 5294.000 5297.000 5300.000 5301.000 5302.000 5303.000 5309.000 5311.000 5317.000
5321.000 5326.000 5328.000 5330.000 5331.000 5343.000 5345.000 5354.000 5355.000 5356.000
5358.000 5368.000 5370.000 5373.000 5389.000 5392.000 5394.000 5402.000 5403.000 5411.000
5412.000 5414.000 5418.000 5425.000 5431.000 5432.000 5433.000 5434.000 5435.000 5437.000
5442.000 5450.000 5451.000 5452.000 5455.000 5456.000 5458.000 5460.000 5466.000 5471.000
5472.000 5490.000 5499.000 5504.000 5505.000 5510.000 5513.000 5521.000 5531.000 5535.000
5537.000 5541.000 5542.000 5543.000 5544.000 5547.000 5551.000 5555.000 5566.000 5568.000
5569.000 5571.000 5572.000 5575.000 5578.000 5580.000 5592.000 5594.000 5595.000 5598.000
5613.000 5614.000 5618.000 5622.000 5725.000 5728.000 5729.000 5731.000 5735.000 5736.000
5738.000 5739.000 5745.000 5749.000 5750.000 5751.000 5760.000 5761.000 5768.000 5769.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

5770.000 5771.000 5772.000 5773.000 5774.000 5777.000 5779.000 5781.000 5783.000 5785.000
5789.000 5794.000 5797.000 5798.000 5799.000 5800.000 5801.000 5802.000 5803.000 5804.000
5805.000 5807.000 5810.000 5812.000 5814.000 5815.000 5816.000 5817.000 5822.000 5824.000
5828.000 5831.000 5835.000 5837.000 5840.000 5843.000 5847.000 5851.000 5853.000 5855.000
5861.000 5865.000 5870.000 5873.000 5875.000 5879.000 5882.000 5883.000 5884.000 5885.000
5886.000 5887.000 5888.000 5889.000 5890.000 5893.000 5898.000 5899.000 5901.000 5903.000
5904.000 5905.000 5906.000 5911.000 5912.000 5916.000 5918.000 5920.000 5923.000 5924.000
5925.000 5926.000 5928.000 5931.000 5934.000 5935.000 5942.000 5946.000 5947.000 5949.000
5953.000 5956.000 5958.000 5965.000 5966.000 5967.000 5968.000 5969.000 5970.000 5971.000
5973.000 5974.000 5977.000 5981.000 5984.000 5987.000 5990.000 5992.000 5993.000 5996.000
5997.000 5998.000 6001.000 6003.000 6006.000 6007.000 6010.000 6011.000 6013.000 6017.000
6021.000 6025.000 6028.000 6029.000 6035.000 6037.000 6038.000 6039.000 6040.000 6041.000
6042.000 6044.000 6045.000 6048.000 6053.000 6055.000 6056.000 6057.000 6059.000 6061.000
6063.000 6065.000 6068.000 6069.000 6073.000 6077.000 6080.000 6082.000 6083.000 6085.000
6088.000 6089.000 6092.000 6094.000 6097.000 6102.000 6105.000 6108.000 6110.000 6111.000
6115.000 6118.000 6120.000 6121.000 6123.000 6125.000 6126.000 6127.000 6128.000 6129.000
6130.000 6134.000 6137.000 6140.000 6144.000 6147.000 6150.000 6153.000 6155.000 6162.000
6166.000 6169.000 6174.000 6178.000 6189.000 6192.000 6195.000 6198.000 6202.000 6205.000
6208.000 6209.000 6210.000 6212.000 6214.000 6216.000 6224.000 6225.000 6227.000 6230.000
6231.000 6232.000 6235.000 6238.000 6239.000 6246.000 6254.000 6257.000 6260.000 6261.000
6263.000 6269.000 6270.000 6276.000 6278.000 6280.000 6282.000 6287.000 6292.000 6295.000
6298.000 6299.000 6302.000 6305.000 6307.000 6308.000 6310.000 6311.000 6316.000 6317.000
6319.000 6328.000 6329.000 6330.000 6332.000 6337.000 6339.000 6340.000 6341.000 6342.000
6344.000 6346.000 6348.000 6352.000 6356.000 6361.000 6364.000 6366.000 6371.000 6372.000
6383.000 6386.000 6394.000 6396.000 6397.000 6398.000 6399.000 6400.000 6401.000 6403.000
6405.000 6407.000 6408.000 6410.000 6412.000 6413.000 6416.000 6417.000 6418.000 6419.000
6420.000 6422.000 6425.000 6427.000 6429.000 6431.000 6433.000 6441.000 6443.000 6450.000
6451.000 6452.000 6465.000 6471.000 6474.000 6476.000 6480.000 6482.000 6485.000 6497.000
6498.000 6501.000 6506.000 6515.000 6516.000 6535.000 6545.000 6546.000 6550.000 6553.000
6567.000 6568.000 6583.000 6587.000 6588.000 6595.000 6609.000 6629.000 6634.000 6638.000
6643.000 6645.000 6653.000 6658.000 6659.000 6664.000 6684.000 6692.000 6694.000 6695.000
6696.000 6703.000 6719.000 6720.000 6729.000 6730.000 6732.000 6734.000 6740.000 6742.000
6745.000 6747.000 6754.000 6755.000 6763.000 6771.000 6774.000 6779.000 6788.000 6791.000
6796.000 6805.000 6806.000 6808.000 6815.000 6816.000 6824.000 6832.000 6835.000 6840.000
6849.000 6852.000 6857.000 6867.000 6868.000 6869.000 6870.000 6878.000 6880.000 6881.000
6889.000 6898.000 6902.000 6908.000 6917.000 6921.000 6927.000 6935.000 6937.000 6938.000
6946.000 6955.000 6959.000 6965.000 6975.000 6979.000 6985.000 6992.000 6993.000 6994.000
7001.000 7011.000 7014.000 7019.000 7028.000 7038.000 7048.000 7049.000 7050.000 7059.000
7065.000 7072.000 7074.000 7078.000 7081.000 7084.000 7090.000 7095.000 7096.000 7104.000
7118.000 7122.000 7128.000 7136.000 7139.000 7140.000 7141.000 7151.000 7157.000 7165.000
7170.000 7176.000 7181.000 7185.000 7201.000 7202.000 7203.000 7204.000 7205.000 7210.000
7234.000 7238.000 7272.000 7273.000 7275.000 7277.000 7278.000 7280.000 7288.000 7290.000
7292.000 7295.000 7298.000 7301.000 7304.000 7307.000 7310.000 7314.000 7317.000 7319.000
7322.000 7325.000 7328.000 7331.000 7339.000 7341.000 7344.000 7345.000 7347.000 7348.000
7355.000 7357.000 7358.000 7381.000 7391.000 7396.000 7402.000 7414.000 7426.000 7428.000
7430.000 7435.000 7437.000 7440.000 7442.000 7453.000 7454.000 7456.000 7460.000 7461.000
7464.000 7468.000 7471.000 7477.000 7479.000 7480.000 7481.000 7486.000 7487.000 7491.000
7492.000 7494.000 7495.000 7498.000 7500.000 7502.000 7505.000 7507.000 7508.000 7509.000
7514.000 7515.000 7517.000 7518.000 7521.000 7524.000 7525.000 7529.000 7531.000 7533.000
7534.000 7539.000 7541.000 7543.000 7549.000 7551.000 7552.000 7553.000 7558.000 7559.000
7563.000 7564.000 7565.000 7567.000 7568.000 7571.000 7573.000 7574.000 7577.000 7578.000
7580.000 7581.000 7584.000 7589.000 7590.000 7592.000 7593.000 7596.000 7599.000 7604.000
7607.000 7609.000 7613.000 7618.000 7619.000 7622.000 7630.000 7639.000 7641.000 7654.000
7655.000 7658.000 7659.000 7661.000 7662.000 7664.000 7666.000 7670.000 7672.000 7673.000
7675.000 7677.000 7685.000 7688.000 7692.000 7695.000 7696.000 7700.000 7702.000 7703.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

7704.000 7714.000 7716.000 7728.000 7730.000 7731.000 7739.000 7740.000 7743.000 7748.000
7749.000 7753.000 7756.000 7759.000 7762.000 7765.000 7766.000 7767.000 7786.000 7801.000
7806.000 7807.000 7808.000 7809.000 7810.000 7813.000 7818.000 7819.000 7822.000 7827.000
7828.000 7829.000 7834.000 7836.000 7840.000 7842.000 7844.000 7846.000 7850.000 7855.000
7856.000 7857.000 7861.000 7865.000 7866.000 7867.000 7868.000 7872.000 7877.000 7878.000
7880.000 7886.000 7891.000 7892.000 7896.000 7897.000 7898.000 7899.000 7903.000 7908.000
7909.000 7911.000 7914.000 7927.000 7929.000 7931.000 7937.000 7938.000 7950.000 7951.000
7953.000 7955.000 7956.000 7958.000 7959.000 7960.000 7961.000 7962.000 7963.000 7965.000
7967.000 7969.000 7977.000 7979.000 7985.000 7989.000 7991.000 7993.000 8007.000 8010.000
8016.000 8019.000 8020.000 8022.000 8023.000 8028.000 8034.000 8039.000 8043.000 8052.000
8053.000 8054.000 8055.000 8059.000 8060.000 8061.000 8069.000

*JUMPAV STRING REFERENCED AT : 6757.000 6818.000 6883.000 6940.000

*JUMPDV STRING REFERENCED AT : 7055.000 7147.000

*JUMPNV STRING REFERENCED AT : 6759.000 6782.000 6820.000 6843.000 6885.000 6911.000 6942.000 6968.000
7005.000 7067.000 7111.000 7159.000

*JUMPS STRING REFERENCED AT : 3069.000 5559.000 5762.000 5959.000 7256.000

*NOP STRING REFERENCED AT : 1753.000 1973.000 2007.000 2395.000 2398.000 4136.000 4612.000 4942.000
5161.000 5488.000 5897.000 6067.000 6268.000 6291.000 6997.000 7027.000 7029.000 7039.000
7100.000 7106.000 7107.000 7109.000 7166.000 7532.000

+ STRING REFERENCED AT : 896.000 934.000 942.000 950.000 960.000 1111.000 1113.000 1118.000
1120.000 1138.000 1185.000 1223.000 1227.000 1258.000 1264.000 1335.000 1356.000 1398.000
1429.000 1454.000 1521.000 1550.000 1551.000 1575.000 1649.000 1765.000 1837.000 1905.000
1928.000 1929.000 1936.000 2265.000 2292.000 2298.000 2364.000 2396.000 2467.000 2493.000
2522.000 2531.000 2689.000 2711.000 2712.000 2970.000 3225.000 3266.000 3271.000 3314.000
3318.000 3321.000 3325.000 3417.000 3450.000 3460.000 3472.000 3474.000 3518.000 3519.000
3533.000 3537.000 3541.000 3543.000 3546.000 3550.000 3551.000 3584.000 3597.000 3604.000
3640.000 3643.000 3670.000 3671.000 3722.000 3763.000 3766.000 3767.000 3772.000 3777.000
3794.000 3796.000 3797.000 3806.000 3808.000 3809.000 3816.000 3817.000 3818.000 3821.000
3836.000 3837.000 3846.000 3848.000 3851.000 3852.000 3876.000 3889.000 3893.000 3894.000
3895.000 3896.000 3920.000 3921.000 3923.000 3924.000 3942.000 3943.000 3961.000 3962.000
3964.000 3965.000 4007.000 4008.000 4010.000 4027.000 4036.000 4060.000 4201.000 4202.000
4229.000 4231.000 4244.000 4246.000 4258.000 4260.000 4292.000 4293.000 4294.000 4295.000
4296.000 4297.000 4298.000 4301.000 4305.000 4306.000 4368.000 4369.000 4370.000 4371.000
4372.000 4373.000 4374.000 4375.000 4377.000 4409.000 4488.000 4537.000 4630.000 4634.000
4764.000 4785.000 4846.000 4869.000 4873.000 4918.000 4941.000 4950.000 4952.000 4962.000
4966.000 4981.000 4982.000 4991.000 4995.000 5013.000 5061.000 5089.000 5094.000 5115.000
5124.000 5127.000 5128.000 5129.000 5132.000 5136.000 5138.000 5140.000 5162.000 5186.000
5191.000 5205.000 5222.000 5223.000 5299.000 5306.000 5456.000 5458.000 5506.000 5508.000
5524.000 5532.000 5547.000 5570.000 5820.000 5919.000 6033.000 6051.000 6075.000 6117.000
6151.000 6306.000 6326.000 6514.000 6543.000 6565.000 6586.000 6589.000 6594.000 6642.000
6643.000 6719.000 6728.000 6753.000 6757.000 6759.000 6767.000 6771.000 6778.000 6779.000
6782.000 6788.000 6795.000 6796.000 6814.000 6828.000 6832.000 6839.000 6840.000 6843.000
6849.000 6856.000 6857.000 6877.000 6883.000 6885.000 6893.000 6897.000 6906.000 6907.000
6911.000 6916.000 6925.000 6926.000 6934.000 6950.000 6954.000 6963.000 6964.000 6968.000
6974.000 6983.000 6984.000 6990.000 6991.000 6996.000 6998.000 7001.000 7002.000 7011.000
7018.000 7019.000 7046.000 7047.000 7052.000 7063.000 7065.000 7078.000 7080.000 7081.000
7091.000 7092.000 7099.000 7101.000 7104.000 7105.000 7117.000 7126.000 7127.000 7135.000
7137.000 7144.000 7155.000 7157.000 7169.000 7175.000 7179.000 7180.000 7205.000 7250.000
7528.000 7597.000 7691.000 8049.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

+1	STRING	REFERENCED AT : 916.000 1396.000 2229.000 2378.000 3680.000 4120.000 5126.000 5489.000 5543.000 5618.000 6464.000 6467.000 6479.000 6627.000 6628.000 7638.000
-	STRING	REFERENCED AT : 1362.000 1365.000 1366.000 1367.000 1437.000 1438.000 1439.000 1442.000 1443.000 1446.000 1447.000 2466.000 2468.000 3167.000 3503.000 3545.000 3574.000 3598.000 3769.000 3770.000 3782.000 3784.000 3850.000 3859.000 3880.000 3905.000 3906.000 3922.000 3930.000 3931.000 4002.000 4003.000 4029.000 4061.000 4084.000 4085.000 4091.000 4092.000 4272.000 4273.000 4515.000 4530.000 4555.000 4727.000 4731.000 4733.000 5245.000 5887.000 6040.000 6129.000 6777.000 6794.000 6818.000 6820.000 6838.000 6855.000 6905.000 6924.000 6940.000 6942.000 6962.000 6982.000 6991.000 6996.000 7003.000 7017.000 7047.000 7052.000 7053.000 7065.000 7067.000 7108.000 7125.000 7137.000 7144.000 7145.000 7157.000 7159.000
-1	STRING	REFERENCED AT : 1305.000 1459.000 2239.000 4590.000 5431.000 7616.000
-A.AND.B	STRING	REFERENCED AT : 8298.000
-A.OR.B	STRING	REFERENCED AT : 8250.000
-B	STRING	REFERENCED AT : 8424.000 8432.000
0	STRING	REFERENCED AT : 6992.000 6999.000 7096.000 7102.000
:	STRING	REFERENCED AT : 51.000 51.000 210.000 210.000 652.000 671.000 719.000 739.000 744.000 749.000 765.000 767.000 770.000 775.000 826.000 835.000 846.000 846.000 868.000 872.000 910.000 927.000 944.000 944.000 949.000 968.000 968.000 984.000 997.000 997.000 997.000 1028.000 1137.000 1167.000 1168.000 1193.000 1194.000 1195.000 1265.000 1276.000 1288.000 1290.000 1298.000 1299.000 1311.000 1315.000 1316.000 1381.000 1387.000 1405.000 1407.000 1458.000 1471.000 1478.000 1479.000 1484.000 1518.000 1533.000 1572.000 1609.000 1611.000 1612.000 1616.000 1616.000 1627.000 1645.000 1632.000 1702.000 1705.000 1748.000 1762.000 1773.000 1802.000 1802.000 1802.000 1810.000 1812.000 1863.000 1866.000 1896.000 1910.000 1910.000 1939.000 1939.000 1952.000 2039.000 2050.000 2050.000 2088.000 2090.000 2100.000 2126.000 2126.000 2132.000 2141.000 2141.000 2151.000 2151.000 2205.000 2207.000 2211.000 2216.000 2218.000 2274.000 2277.000 2371.000 2372.000 2379.000 2400.000 2402.000 2409.000 2428.000 2429.000 2472.000 2474.000 2576.000 2597.000 2609.000 2629.000 2634.000 2639.000 2660.000 2674.000 2675.000 2679.000 2681.000 2681.000 2771.000 2774.000 2777.000 2808.000 2828.000 2837.000 2841.000 2863.000 2872.000 2898.000 2906.000 2918.000 2931.000 2940.000 2943.000 2944.000 2946.000 2955.000 3010.000 3066.000 3097.000 3135.000 3155.000 3173.000 3180.000 3206.000 3209.000 3214.000 3227.000 3227.000 3242.000 3245.000 3273.000 3305.000 3308.000 3309.000 3349.000 3415.000 3419.000 3419.000 3422.000 3423.000 3521.000 3521.000 3564.000 3570.000 3603.000 3790.000 3925.000 3926.000 3995.000 3999.000 4011.000 4012.000 4249.000 4265.000 4405.000 4410.000 4578.000 4636.000 4636.000 4713.000 4781.000 4785.000 4785.000 4817.000 4817.000 4826.000 4826.000 4834.000 4834.000 4848.000 4848.000 4879.000 4879.000 4896.000 4896.000 4911.000 4918.000 4918.000 4939.000 4943.000 4943.000 5116.000 5117.000 5193.000 5193.000 5247.000 5259.000 5275.000 5277.000 5280.000 5301.000 5301.000 5305.000 5309.000 5309.000 5319.000 5365.000 5367.000 5429.000 5466.000 5466.000 5531.000 5531.000 5574.000 5606.000 5839.000 5843.000 5844.000 5867.000 5877.000 5912.000 5912.000 5915.000 5941.000 5988.000 5999.000 6016.000 6050.000 6054.000 6098.000 6103.000 6171.000 6172.000 6185.000 6187.000 6200.000 6220.000 6232.000 6240.000 6251.000 6262.000 6334.000 6342.000 6353.000 6372.000 6399.000 6405.000 6417.000 6417.000 6497.000 6497.000 6567.000 6567.000 6587.000 6587.000 6654.000 6665.000 6672.000 6677.000 6684.000 6694.000 6745.000 6745.000 7210.000 7244.000 7271.000 7275.000 7294.000 7297.000 7300.000 7303.000 7306.000 7309.000 7312.000 7316.000 7324.000 7325.000 7328.000 7330.000 7333.000 7425.000 7428.000 7430.000 7439.000 7471.000 7492.000 7564.000 7590.000 7656.000 7663.000 7669.000 7690.000 7745.000 7759.000 7821.000 7827.000 7828.000 7628.000 7860.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

7885.000 7902.000 7913.000 7950.000 7955.000 7955.000 7955.000 7963.000 7965.000 7967.000
7969.000 7984.000 7990.000 7992.000 8034.000 8041.000 8056.000 8059.000 8059.000 8059.000

:% STRING REFERENCED AT : 1702.000 4777.000 4906.000 5022.000 5036.000

= STRING REFERENCED AT : 94.000 94.000 410.000 594.000 597.000 603.000 639.000 652.000
672.000 674.000 676.000 677.000 678.000 682.000 692.000 694.000 695.000 698.000
699.000 705.000 714.000 719.000 739.000 744.000 767.000 768.000 770.000 775.000
785.000 786.000 787.000 788.000 800.000 801.000 812.000 814.000 816.000 817.000
818.000 820.000 821.000 822.000 824.000 835.000 844.000 845.000 846.000 847.000
858.000 859.000 860.000 864.000 865.000 866.000 868.000 869.000 870.000 871.000
872.000 884.000 896.000 897.000 905.000 906.000 906.000 908.000 909.000 910.000
911.000 913.000 914.000 915.000 916.000 917.000 919.000 927.000 928.000 929.000
932.000 934.000 936.000 937.000 941.000 942.000 943.000 946.000 947.000 948.000
949.000 950.000 951.000 952.000 953.000 955.000 957.000 959.000 960.000 961.000
963.000 964.000 965.000 966.000 969.000 970.000 984.000 996.000 997.000 998.000
1002.000 1004.000 1005.000 1009.000 1011.000 1012.000 1013.000 1015.000 1016.000 1024.000
1026.000 1027.000 1028.000 1036.000 1037.000 1044.000 1045.000 1046.000 1047.000 1050.000
1056.000 1059.000 1062.000 1064.000 1072.000 1074.000 1083.000 1084.000 1085.000 1086.000
1087.000 1088.000 1089.000 1092.000 1096.000 1097.000 1101.000 1102.000 1103.000 1104.000
1105.000 1107.000 1110.000 1111.000 1112.000 1113.000 1114.000 1118.000 1119.000 1120.000
1121.000 1124.000 1125.000 1128.000 1135.000 1137.000 1138.000 1150.000 1151.000 1154.000
1156.000 1165.000 1167.000 1168.000 1169.000 1181.000 1182.000 1184.000 1185.000 1186.000
1192.000 1193.000 1194.000 1195.000 1196.000 1197.000 1203.000 1204.000 1205.000 1207.000
1208.000 1215.000 1216.000 1217.000 1219.000 1220.000 1221.000 1222.000 1223.000 1224.000
1225.000 1227.000 1228.000 1230.000 1231.000 1232.000 1233.000 1235.000 1236.000 1254.000
1255.000 1257.000 1258.000 1259.000 1263.000 1264.000 1265.000 1268.000 1269.000 1270.000
1274.000 1275.000 1276.000 1277.000 1279.000 1281.000 1283.000 1284.000 1285.000 1286.000
1287.000 1288.000 1289.000 1290.000 1291.000 1292.000 1294.000 1295.000 1297.000 1298.000
1299.000 1300.000 1301.000 1302.000 1305.000 1309.000 1310.000 1316.000 1319.000 1322.000
1323.000 1327.000 1332.000 1333.000 1334.000 1335.000 1336.000 1337.000 1338.000 1340.000
1342.000 1344.000 1345.000 1346.000 1352.000 1353.000 1355.000 1356.000 1358.000 1361.000
1362.000 1363.000 1365.000 1366.000 1367.000 1368.000 1370.000 1374.000 1375.000 1380.000
1381.000 1382.000 1384.000 1385.000 1386.000 1387.000 1389.000 1390.000 1391.000 1396.000
1397.000 1398.000 1399.000 1404.000 1405.000 1406.000 1407.000 1408.000 1424.000 1425.000
1426.000 1427.000 1429.000 1431.000 1433.000 1434.000 1435.000 1436.000 1437.000 1438.000
1439.000 1442.000 1443.000 1446.000 1447.000 1450.000 1453.000 1454.000 1456.000 1458.000
1459.000 1461.000 1464.000 1465.000 1466.000 1467.000 1468.000 1469.000 1470.000 1471.000
1478.000 1479.000 1484.000 1513.000 1515.000 1516.000 1517.000 1520.000 1521.000 1522.000
1532.000 1533.000 1549.000 1550.000 1551.000 1556.000 1557.000 1570.000 1571.000 1572.000
1573.000 1574.000 1575.000 1576.000 1583.000 1585.000 1593.000 1594.000 1606.000 1607.000
1608.000 1609.000 1610.000 1611.000 1612.000 1616.000 1620.000 1624.000 1625.000 1642.000
1643.000 1644.000 1648.000 1649.000 1650.000 1656.000 1664.000 1665.000 1675.000 1676.000
1677.000 1678.000 1681.000 1682.000 1686.000 1701.000 1702.000 1704.000 1705.000 1713.000
1714.000 1715.000 1716.000 1723.000 1724.000 1725.000 1726.000 1731.000 1734.000 1737.000
1738.000 1739.000 1740.000 1746.000 1748.000 1752.000 1755.000 1756.000 1762.000 1765.000
1767.000 1768.000 1772.000 1773.000 1774.000 1775.000 1779.000 1780.000 1781.000 1783.000
1784.000 1785.000 1796.000 1807.000 1810.000 1811.000 1812.000 1813.000 1814.000 1815.000
1816.000 1817.000 1818.000 1819.000 1821.000 1835.000 1837.000 1838.000 1839.000 1840.000
1842.000 1844.000 1849.000 1859.000 1863.000 1865.000 1866.000 1890.000 1891.000 1893.000
1894.000 1895.000 1896.000 1897.000 1898.000 1899.000 1900.000 1901.000 1904.000 1905.000
1906.000 1908.000 1911.000 1928.000 1929.000 1930.000 1930.000 1931.000 1932.000 1934.000
1935.000 1936.000 1937.000 1940.000 1951.000 1952.000 1953.000 2027.000 2028.000 2039.000
2040.000 2041.000 2042.000 2043.000 2045.000 2045.000 2049.000 2053.000 2054.000 2055.000
2058.000 2065.000 2066.000 2067.000 2068.000 2070.000 2076.000 2077.000 2086.000 2087.000

SEL 32/75 CPU

MICROCODE CROSS-REFERENCE LIST

2088.000 2089.000 2090.000 2092.000 2100.000 2117.000 2118.000 2121.000 2122.000 2124.000
2125.000 2127.000 2140.000 2141.000 2142.000 2143.000 2144.000 2145.000 2152.000 2153.000
2161.000 2163.000 2166.000 2167.000 2179.000 2180.000 2181.000 2183.000 2184.000 2192.000
2193.000 2201.000 2202.000 2204.000 2206.000 2217.000 2222.000 2223.000 2224.000 2226.000
2228.000 2229.000 2235.000 2237.000 2238.000 2239.000 2243.000 2244.000 2245.000 2248.000
2249.000 2255.000 2256.000 2257.000 2261.000 2262.000 2263.000 2264.000 2265.000 2266.000
2267.000 2268.000 2270.000 2271.000 2272.000 2274.000 2275.000 2289.000 2290.000 2291.000
2292.000 2293.000 2294.000 2296.000 2297.000 2298.000 2299.000 2300.000 2301.000 2308.000
2309.000 2310.000 2311.000 2315.000 2316.000 2317.000 2324.000 2325.000 2327.000 2328.000
2331.000 2340.000 2341.000 2342.000 2343.000 2351.000 2362.000 2363.000 2364.000 2369.000
2370.000 2371.000 2372.000 2377.000 2378.000 2379.000 2381.000 2394.000 2396.000 2397.000
2400.000 2401.000 2402.000 2408.000 2411.000 2422.000 2423.000 2424.000 2426.000 2435.000
2446.000 2447.000 2448.000 2455.000 2460.000 2461.000 2462.000 2464.000 2465.000 2466.000
2467.000 2468.000 2470.000 2471.000 2472.000 2473.000 2474.000 2476.000 2477.000 2493.000
2495.000 2496.000 2497.000 2498.000 2499.000 2501.000 2516.000 2520.000 2522.000 2524.000
2526.000 2527.000 2529.000 2530.000 2531.000 2532.000 2533.000 2535.000 2536.000 2537.000
2538.000 2540.000 2543.000 2548.000 2550.000 2551.000 2552.000 2560.000 2561.000 2566.000
2573.000 2575.000 2576.000 2578.000 2584.000 2586.000 2587.000 2593.000 2596.000 2597.000
2599.000 2605.000 2607.000 2608.000 2609.000 2610.000 2618.000 2620.000 2621.000 2622.000
2629.000 2634.000 2637.000 2639.000 2640.000 2660.000 2661.000 2662.000 2663.000 2664.000
2665.000 2666.000 2667.000 2668.000 2672.000 2674.000 2675.000 2676.000 2677.000 2678.000
2679.000 2682.000 2684.000 2686.000 2688.000 2689.000 2690.000 2691.000 2693.000 2695.000
2696.000 2699.000 2709.000 2710.000 2711.000 2712.000 2713.000 2718.000 2719.000 2719.000
2720.000 2721.000 2727.000 2728.000 2732.000 2733.000 2734.000 2735.000 2737.000 2740.000
2741.000 2742.000 2743.000 2744.000 2745.000 2746.000 2747.000 2748.000 2755.000 2763.000
2764.000 2765.000 2766.000 2767.000 2768.000 2769.000 2771.000 2772.000 2774.000 2775.000
2776.000 2777.000 2778.000 2782.000 2789.000 2805.000 2806.000 2808.000 2815.000 2824.000
2825.000 2826.000 2828.000 2834.000 2835.000 2837.000 2838.000 2839.000 2841.000 2842.000
2850.000 2859.000 2860.000 2861.000 2863.000 2870.000 2871.000 2872.000 2880.000 2881.000
2886.000 2887.000 2888.000 2890.000 2897.000 2898.000 2905.000 2906.000 2914.000 2915.000
2916.000 2917.000 2918.000 2926.000 2927.000 2929.000 2930.000 2931.000 2937.000 2938.000
2939.000 2940.000 2941.000 2942.000 2943.000 2944.000 2945.000 2946.000 2955.000 2956.000
2957.000 2958.000 2965.000 2966.000 2967.000 2968.000 2969.000 2970.000 2971.000 2981.000
2997.000 3000.000 3003.000 3006.000 3008.000 3010.000 3025.000 3027.000 3030.000 3033.000
3034.000 3054.000 3057.000 3060.000 3063.000 3065.000 3066.000 3081.000 3095.000 3097.000
3098.000 3100.000 3102.000 3107.000 3108.000 3110.000 3115.000 3117.000 3119.000 3121.000
3124.000 3126.000 3132.000 3133.000 3134.000 3135.000 3141.000 3146.000 3147.000 3152.000
3153.000 3155.000 3156.000 3159.000 3160.000 3162.000 3164.000 3166.000 3167.000 3170.000
3171.000 3172.000 3173.000 3174.000 3178.000 3179.000 3180.000 3204.000 3205.000 3206.000
3209.000 3210.000 3211.000 3213.000 3214.000 3215.000 3216.000 3217.000 3218.000 3220.000
3221.000 3224.000 3225.000 3226.000 3229.000 3230.000 3231.000 3241.000 3242.000 3243.000
3245.000 3246.000 3266.000 3271.000 3273.000 3304.000 3305.000 3306.000 3307.000 3308.000
3309.000 3313.000 3314.000 3315.000 3316.000 3317.000 3318.000 3319.000 3320.000 3321.000
3322.000 3323.000 3324.000 3325.000 3326.000 3347.000 3348.000 3349.000 3350.000 3351.000
3354.000 3357.000 3382.000 3387.000 3388.000 3389.000 3390.000 3406.000 3411.000 3413.000
3414.000 3415.000 3417.000 3418.000 3421.000 3422.000 3423.000 3424.000 3427.000 3442.000
3443.000 3450.000 3451.000 3452.000 3460.000 3461.000 3463.000 3464.000 3472.000 3474.000
3476.000 3479.000 3502.000 3503.000 3516.000 3518.000 3519.000 3522.000 3527.000 3529.000
3530.000 3531.000 3533.000 3536.000 3537.000 3540.000 3541.000 3542.000 3543.000 3545.000
3546.000 3547.000 3548.000 3550.000 3551.000 3554.000 3557.000 3560.000 3563.000 3564.000
3570.000 3572.000 3574.000 3582.000 3583.000 3584.000 3585.000 3587.000 3595.000 3597.000
3598.000 3599.000 3603.000 3604.000 3610.000 3611.000 3626.000 3627.000 3628.000 3629.000
3630.000 3631.000 3633.000 3635.000 3640.000 3641.000 3643.000 3644.000 3645.000 3646.000
3648.000 3651.000 3660.000 3670.000 3671.000 3675.000 3680.000 3683.000 3687.000 3690.000
3691.000 3694.000 3696.000 3699.000 3702.000 3705.000 3706.000 3708.000 3711.000 3714.000

SEL 32/75 CPU

MICROCODE CROSS-REFFRENCE LIST

3716.000 3719.000 3720.000 3722.000 3730.000 3731.000 3736.000 3737.000 3738.000 3743.000
3744.000 3745.000 3750.000 3751.000 3752.000 3757.000 3758.000 3762.000 3763.000 3764.000
3765.000 3766.000 3767.000 3768.000 3769.000 3770.000 3771.000 3772.000 3773.000 3777.000
3778.000 3780.000 3781.000 3782.000 3783.000 3784.000 3785.000 3788.000 3789.000 3790.000
3794.000 3795.000 3796.000 3797.000 3802.000 3803.000 3804.000 3805.000 3806.000 3807.000
3808.000 3809.000 3811.000 3812.000 3813.000 3814.000 3816.000 3817.000 3818.000 3821.000
3822.000 3823.000 3826.000 3831.000 3832.000 3834.000 3835.000 3836.000 3837.000 3841.000
3846.000 3847.000 3848.000 3849.000 3850.000 3851.000 3852.000 3853.000 3854.000 3855.000
3856.000 3857.000 3858.000 3859.000 3876.000 3877.000 3878.000 3879.000 3880.000 3881.000
3889.000 3891.000 3893.000 3894.000 3895.000 3896.000 3897.000 3898.000 3901.000 3904.000
3905.000 3906.000 3907.000 3910.000 3911.000 3912.000 3913.000 3914.000 3915.000 3916.000
3917.000 3918.000 3919.000 3920.000 3921.000 3922.000 3923.000 3924.000 3925.000 3926.000
3927.000 3928.000 3929.000 3930.000 3931.000 3932.000 3934.000 3935.000 3936.000 3937.000
3938.000 3939.000 3942.000 3943.000 3961.000 3962.000 3963.000 3964.000 3965.000 3966.000
3968.000 3973.000 3975.000 3976.000 3977.000 3981.000 3985.000 3987.000 3990.000 3991.000
3992.000 3993.000 3994.000 3995.000 3996.000 3997.000 3998.000 3999.000 4000.000 4001.000
4002.000 4003.000 4004.000 4007.000 4008.000 4009.000 4010.000 4011.000 4012.000 4013.000
4024.000 4025.000 4026.000 4027.000 4028.000 4029.000 4031.000 4035.000 4036.000 4038.000
4040.000 4041.000 4042.000 4044.000 4045.000 4046.000 4050.000 4055.000 4060.000 4061.000
4063.000 4063.000 4064.000 4065.000 4066.000 4067.000 4068.000 4073.000 4075.000 4076.000
4077.000 4078.000 4080.000 4084.000 4085.000 4091.000 4092.000 4097.000 4103.000 4107.000
4111.000 4118.000 4120.000 4122.000 4124.000 4131.000 4133.000 4138.000 4140.000 4141.000
4145.000 4147.000 4148.000 4149.000 4150.000 4151.000 4152.000 4153.000 4183.000 4184.000
4185.000 4190.000 4191.000 4192.000 4193.000 4195.000 4196.000 4197.000 4201.000 4202.000
4205.000 4206.000 4207.000 4208.000 4209.000 4210.000 4214.000 4215.000 4216.000 4217.000
4218.000 4219.000 4220.000 4221.000 4222.000 4226.000 4228.000 4229.000 4231.000 4233.000
4234.000 4235.000 4240.000 4242.000 4244.000 4245.000 4246.000 4248.000 4249.000 4250.000
4251.000 4255.000 4257.000 4258.000 4260.000 4262.000 4263.000 4264.000 4265.000 4266.000
4267.000 4269.000 4270.000 4271.000 4272.000 4273.000 4276.000 4278.000 4284.000 4285.000
4288.000 4289.000 4292.000 4293.000 4294.000 4295.000 4296.000 4297.000 4298.000 4301.000
4302.000 4305.000 4306.000 4368.000 4369.000 4370.000 4371.000 4372.000 4373.000 4374.000
4375.000 4377.000 4378.000 4379.000 4380.000 4404.000 4405.000 4409.000 4410.000 4422.000
4429.000 4442.000 4457.000 4459.000 4464.000 4468.000 4471.000 4472.000 4473.000 4475.000
4476.000 4480.000 4482.000 4486.000 4487.000 4488.000 4492.000 4493.000 4497.000 4498.000
4499.000 4500.000 4507.000 4508.000 4511.000 4514.000 4515.000 4516.000 4517.000 4518.000
4519.000 4520.000 4530.000 4531.000 4534.000 4537.000 4539.000 4544.000 4545.000 4550.000
4553.000 4555.000 4558.000 4560.000 4566.000 4568.000 4569.000 4571.000 4572.000 4576.000
4577.000 4578.000 4583.000 4584.000 4585.000 4586.000 4587.000 4589.000 4590.000 4591.000
4594.000 4595.000 4596.000 4597.000 4598.000 4602.000 4605.000 4607.000 4608.000 4609.000
4610.000 4611.000 4614.000 4615.000 4616.000 4617.000 4618.000 4619.000 4630.000 4634.000
4637.000 4649.000 4655.000 4657.000 4661.000 4670.000 4674.000 4687.000 4695.000 4704.000
4708.000 4712.000 4713.000 4715.000 4727.000 4731.000 4733.000 4741.000 4744.000 4746.000
4748.000 4752.000 4760.000 4761.000 4762.000 4763.000 4764.000 4766.000 4767.000 4768.000
4776.000 4777.000 4778.000 4779.000 4781.000 4782.000 4784.000 4785.000 4789.000 4790.000
4796.000 4813.000 4814.000 4816.000 4818.000 4822.000 4823.000 4825.000 4827.000 4830.000
4831.000 4833.000 4835.000 4843.000 4846.000 4847.000 4849.000 4861.000 4869.000 4870.000
4871.000 4873.000 4877.000 4880.000 4881.000 4895.000 4897.000 4906.000 4907.000 4908.000
4909.000 4911.000 4912.000 4914.000 4918.000 4922.000 4923.000 4939.000 4941.000 4944.000
4945.000 4950.000 4952.000 4962.000 4964.000 4966.000 4969.000 4970.000 4975.000 4977.000
4981.000 4982.000 4991.000 4993.000 4995.000 4998.000 4999.000 5004.000 5013.000 5022.000
5023.000 5025.000 5026.000 5027.000 5036.000 5037.000 5038.000 5039.000 5040.000 5048.000
5052.000 5061.000 5069.000 5077.000 5089.000 5094.000 5111.000 5112.000 5114.000 5115.000
5116.000 5117.000 5124.000 5125.000 5126.000 5127.000 5128.000 5129.000 5131.000 5132.000
5133.000 5134.000 5135.000 5136.000 5138.000 5140.000 5144.000 5146.000 5154.000 5155.000
5156.000 5157.000 5158.000 5160.000 5162.000 5170.000 5173.000 5174.000 5175.000 5177.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

5185.000 5186.000 5191.000 5194.000 5196.000 5205.000 5206.000 5211.000 5219.000 5222.000
5223.000 5224.000 5233.000 5240.000 5242.000 5243.000 5244.000 5245.000 5246.000 5247.000
5249.000 5250.000 5251.000 5252.000 5255.000 5257.000 5259.000 5268.000 5271.000 5275.000
5277.000 5279.000 5280.000 5299.000 5305.000 5306.000 5307.000 5308.000 5309.000 5311.000
5314.000 5317.000 5319.000 5321.000 5323.000 5324.000 5329.000 5330.000 5343.000 5345.000
5346.000 5347.000 5349.000 5351.000 5353.000 5354.000 5360.000 5362.000 5363.000 5364.000
5365.000 5366.000 5367.000 5368.000 5403.000 5411.000 5412.000 5416.000 5429.000 5431.000
5439.000 5450.000 5451.000 5452.000 5455.000 5456.000 5457.000 5458.000 5462.000 5464.000
5466.000 5487.000 5489.000 5499.000 5500.000 5501.000 5502.000 5503.000 5506.000 5507.000
5508.000 5509.000 5510.000 5513.000 5521.000 5522.000 5524.000 5525.000 5531.000 5532.000
5533.000 5543.000 5544.000 5545.000 5546.000 5547.000 5550.000 5553.000 5566.000 5570.000
5571.000 5574.000 5593.000 5594.000 5598.000 5602.000 5606.000 5612.000 5613.000 5617.000
5618.000 5726.000 5727.000 5730.000 5731.000 5746.000 5747.000 5748.000 5749.000 5757.000
5759.000 5759.000 5761.000 5777.000 5779.000 5781.000 5783.000 5785.000 5787.000 5789.000
5791.000 5792.000 5793.000 5794.000 5798.000 5801.000 5803.000 5811.000 5812.000 5814.000
5815.000 5816.000 5818.000 5819.000 5820.000 5821.000 5826.000 5827.000 5828.000 5832.000
5833.000 5835.000 5838.000 5839.000 5840.000 5842.000 5843.000 5844.000 5845.000 5850.000
5851.000 5856.000 5857.000 5858.000 5860.000 5866.000 5867.000 5868.000 5870.000 5871.000
5872.000 5876.000 5877.000 5878.000 5887.000 5899.000 5901.000 5903.000 5904.000 5906.000
5907.000 5911.000 5913.000 5914.000 5915.000 5916.000 5919.000 5920.000 5927.000 5930.000
5931.000 5932.000 5933.000 5934.000 5935.000 5936.000 5938.000 5939.000 5940.000 5941.000
5942.000 5944.000 5945.000 5946.000 5949.000 5954.000 5955.000 5957.000 5958.000 5975.000
5977.000 5978.000 5979.000 5982.000 5983.000 5984.000 5985.000 5986.000 5988.000 5989.000
5994.000 5995.000 5996.000 5997.000 5999.000 6000.000 6003.000 6008.000 6012.000 6014.000
6015.000 6016.000 6018.000 6019.000 6020.000 6024.000 6025.000 6026.000 6027.000 6028.000
6029.000 6031.000 6033.000 6034.000 6037.000 6038.000 6040.000 6046.000 6049.000 6050.000
6051.000 6052.000 6054.000 6056.000 6062.000 6063.000 6066.000 6070.000 6071.000 6072.000
6074.000 6075.000 6076.000 6078.000 6079.000 6090.000 6091.000 6092.000 6095.000 6096.000
6097.000 6098.000 6099.000 6101.000 6103.000 6104.000 6106.000 6107.000 6109.000 6114.000
6115.000 6116.000 6117.000 6122.000 6129.000 6132.000 6133.000 6135.000 6136.000 6139.000
6140.000 6141.000 6142.000 6143.000 6148.000 6149.000 6150.000 6151.000 6152.000 6155.000
6157.000 6158.000 6159.000 6160.000 6161.000 6164.000 6165.000 6167.000 6168.000 6170.000
6171.000 6172.000 6173.000 6176.000 6177.000 6179.000 6180.000 6181.000 6182.000 6183.000
6184.000 6185.000 6186.000 6187.000 6188.000 6193.000 6194.000 6196.000 6197.000 6199.000
6200.000 6201.000 6206.000 6207.000 6209.000 6211.000 6219.000 6220.000 6221.000 6222.000
6223.000 6225.000 6228.000 6229.000 6232.000 6236.000 6237.000 6240.000 6242.000 6243.000
6244.000 6245.000 6248.000 6249.000 6250.000 6251.000 6252.000 6253.000 6258.000 6259.000
6262.000 6266.000 6267.000 6271.000 6272.000 6273.000 6274.000 6275.000 6276.000 6278.000
6280.000 6282.000 6284.000 6286.000 6290.000 6293.000 6294.000 6296.000 6297.000 6301.000
6306.000 6311.000 6313.000 6314.000 6318.000 6321.000 6322.000 6323.000 6324.000 6325.000
6326.000 6327.000 6333.000 6334.000 6335.000 6336.000 6342.000 6344.000 6346.000 6348.000
6350.000 6351.000 6353.000 6354.000 6355.000 6359.000 6360.000 6364.000 6366.000 6368.000
6369.000 6372.000 6382.000 6383.000 6384.000 6385.000 6386.000 6389.000 6390.000 6390.000
6394.000 6398.000 6405.000 6415.000 6416.000 6417.000 6418.000 6424.000 6432.000 6442.000
6464.000 6465.000 6466.000 6467.000 6468.000 6471.000 6474.000 6476.000 6479.000 6480.000
6481.000 6483.000 6484.000 6485.000 6498.000 6499.000 6500.000 6501.000 6505.000 6514.000
6515.000 6516.000 6533.000 6534.000 6535.000 6540.000 6541.000 6543.000 6545.000 6546.000
6548.000 6549.000 6550.000 6552.000 6564.000 6565.000 6568.000 6585.000 6586.000 6588.000
6589.000 6590.000 6594.000 6595.000 6599.000 6608.000 6609.000 6625.000 6626.000 6627.000
6628.000 6629.000 6633.000 6634.000 6642.000 6643.000 6654.000 6658.000 6660.000 6661.000
6663.000 6664.000 6665.000 6666.000 6667.000 6668.000 6669.000 6670.000 6671.000 6672.000
6674.000 6675.000 6676.000 6677.000 6680.000 6682.000 6684.000 6695.000 6701.000 6702.000
6719.000 6727.000 6728.000 6730.000 6731.000 6741.000 6752.000 6753.000 6753.000 6756.000
6757.000 6759.000 6759.000 6762.000 6767.000 6767.000 6771.000 6771.000 6774.000 6777.000
6777.000 6778.000 6779.000 6779.000 6782.000 6782.000 6785.000 6788.000 6788.000 6791.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

6794.000 6794.000 6795.000 6796.000 6796.000 6807.000 6813.000 6814.000 6814.000 6817.000
6818.000 6820.000 6820.000 6823.000 6828.000 6828.000 6832.000 6832.000 6835.000 6838.000
6838.000 6839.000 6840.000 6840.000 6843.000 6843.000 6846.000 6849.000 6849.000 6852.000
6855.000 6855.000 6856.000 6857.000 6857.000 6868.000 6875.000 6876.000 6877.000 6877.000
6878.000 6880.000 6882.000 6883.000 6885.000 6885.000 6888.000 6893.000 6893.000 6897.000
6897.000 6898.000 6901.000 6902.000 6905.000 6905.000 6906.000 6907.000 6907.000 6908.000
6911.000 6913.000 6916.000 6916.000 6917.000 6920.000 6921.000 6924.000 6924.000 6925.000
6926.000 6926.000 6927.000 6932.000 6933.000 6934.000 6934.000 6935.000 6937.000 6939.000
6940.000 6942.000 6942.000 6945.000 6950.000 6950.000 6954.000 6954.000 6955.000 6958.000
6959.000 6962.000 6962.000 6963.000 6964.000 6964.000 6965.000 6968.000 6971.000 6974.000
6974.000 6975.000 6978.000 6979.000 6982.000 6982.000 6983.000 6984.000 6984.000 6985.000
6989.000 6990.000 6990.000 6991.000 6991.000 6992.000 6993.000 6993.000 6995.000 6995.000
6996.000 6996.000 6998.000 6998.000 6999.000 7001.000 7002.000 7003.000 7004.000 7005.000
7008.000 7011.000 7011.000 7014.000 7017.000 7017.000 7018.000 7018.000 7019.000 7019.000
7026.000 7036.000 7045.000 7046.000 7046.000 7047.000 7047.000 7049.000 7049.000 7051.000
7051.000 7052.000 7052.000 7053.000 7054.000 7055.000 7057.000 7059.000 7061.000 7063.000
7063.000 7065.000 7065.000 7066.000 7067.000 7070.000 7074.000 7076.000 7078.000 7078.000
7080.000 7081.000 7081.000 7083.000 7084.000 7089.000 7090.000 7091.000 7091.000 7092.000
7092.000 7096.000 7097.000 7098.000 7098.000 7099.000 7101.000 7101.000 7102.000 7104.000
7105.000 7108.000 7110.000 7114.000 7117.000 7117.000 7118.000 7121.000 7122.000 7125.000
7125.000 7126.000 7127.000 7127.000 7128.000 7133.000 7134.000 7135.000 7135.000 7136.000
7137.000 7137.000 7137.000 7139.000 7139.000 7142.000 7143.000 7143.000 7144.000 7144.000
7145.000 7146.000 7147.000 7149.000 7151.000 7153.000 7155.000 7155.000 7157.000 7157.000
7158.000 7159.000 7162.000 7169.000 7173.000 7175.000 7175.000 7176.000 7179.000 7180.000
7180.000 7181.000 7183.000 7184.000 7185.000 7201.000 7205.000 7206.000 7207.000 7209.000
7210.000 7232.000 7233.000 7239.000 7243.000 7244.000 7245.000 7246.000 7250.000 7255.000
7256.000 7270.000 7271.000 7273.000 7274.000 7275.000 7279.000 7280.000 7294.000 7295.000
7297.000 7298.000 7300.000 7301.000 7303.000 7306.000 7307.000 7309.000 7310.000 7312.000
7316.000 7317.000 7321.000 7324.000 7325.000 7327.000 7328.000 7330.000 7331.000 7333.000
7335.000 7337.000 7339.000 7340.000 7341.000 7344.000 7347.000 7348.000 7349.000 7351.000
7352.000 7354.000 7381.000 7387.000 7390.000 7396.000 7402.000 7408.000 7410.000 7412.000
7414.000 7423.000 7424.000 7425.000 7426.000 7428.000 7430.000 7434.000 7439.000 7440.000
7441.000 7442.000 7451.000 7452.000 7453.000 7454.000 7456.000 7459.000 7460.000 7461.000
7462.000 7466.000 7467.000 7468.000 7470.000 7471.000 7475.000 7477.000 7481.000 7483.000
7484.000 7487.000 7488.000 7491.000 7496.000 7502.000 7503.000 7509.000 7512.000 7514.000
7519.000 7525.000 7526.000 7527.000 7528.000 7529.000 7530.000 7533.000 7534.000 7536.000
7537.000 7538.000 7540.000 7547.000 7549.000 7553.000 7555.000 7556.000 7559.000 7560.000
7563.000 7569.000 7574.000 7575.000 7582.000 7587.000 7589.000 7596.000 7597.000 7598.000
7599.000 7600.000 7601.000 7602.000 7605.000 7609.000 7615.000 7616.000 7617.000 7618.000
7619.000 7620.000 7622.000 7626.000 7627.000 7629.000 7630.000 7637.000 7638.000 7639.000
7640.000 7641.000 7650.000 7651.000 7652.000 7653.000 7656.000 7657.000 7658.000 7661.000
7663.000 7666.000 7668.000 7669.000 7672.000 7674.000 7677.000 7682.000 7683.000 7684.000
7685.000 7686.000 7687.000 7689.000 7690.000 7691.000 7693.000 7694.000 7696.000 7699.000
7701.000 7713.000 7715.000 7716.000 7725.000 7726.000 7729.000 7731.000 7738.000 7740.000
7742.000 7744.000 7745.000 7746.000 7747.000 7750.000 7751.000 7752.000 7756.000 7758.000
7759.000 7784.000 7785.000 7799.000 7800.000 7805.000 7810.000 7811.000 7815.000 7816.000
7817.000 7818.000 7820.000 7821.000 7828.000 7829.000 7831.000 7832.000 7833.000 7837.000
7838.000 7839.000 7846.000 7848.000 7856.000 7859.000 7860.000 7865.000 7866.000 7879.000
7881.000 7882.000 7883.000 7884.000 7885.000 7891.000 7892.000 7896.000 7901.000 7902.000
7910.000 7912.000 7913.000 7952.000 7954.000 7955.000 7958.000 7959.000 7963.000 7965.000
7967.000 7969.000 7978.000 7980.000 7981.000 7982.000 7983.000 7984.000 7985.000 7986.000
7987.000 7988.000 7989.000 7990.000 7992.000 7993.000 8008.000 8010.000 8014.000 8016.000
8017.000 8021.000 8029.000 8030.000 8031.000 8032.000 8033.000 8034.000 8035.000 8036.000
8037.000 8041.000 8043.000 8046.000 8048.000 8049.000 8052.000 8054.000 8056.000 8058.000
8059.000 8061.000 8069.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
A                       (P)FIELD REFERENCED AT :    51.000

A.AND.B                 STRING   REFERENCED AT :  8251.000 8292.000 8300.000 8303.000 8320.000 8491.000

A.EOR.B                 STRING   REFERENCED AT :  8294.000 8296.000 8493.000 8495.000 8501.000

A.MINUS.B               STRING   REFERENCED AT :  8295.000 8304.000 8306.000 8352.000 8400.000 8425.000 8433.000 8494.000
                                 8497.000 8505.000

A.OR.B                  STRING   REFERENCED AT :  8293.000 8297.000 8492.000 8500.000

A.PLUS.B                STRING   REFERENCED AT :  8299.000 8305.000 8324.000 8360.000 8399.000 8496.000 8504.000

ABM                     LABEL    REFERENCED AT :  4980.000 8299.000

ABM.EXIT                LABEL    REFERENCED AT :  4978.000 4982.000

ABR                     LABEL    REFERENCED AT :  4537.000 8252.000

ABS.OPR                 LABEL    REFERENCED AT :  3913.000 4006.000 4185.000 4198.000

ABSDI                   STRING   REFERENCED AT :  3543.000 3597.000 3640.000 3763.000 3821.000 3848.000 3851.000 3893.000
                                 3896.000 3920.000 3921.000 3924.000 4007.000 4008.000 4010.000

ABST                    STRING   REFERENCED AT :  2970.000 3518.000 3541.000 3584.000 3670.000 3766.000 3767.000 3794.000
                                 3806.000 3846.000 3889.000 4027.000

ACCUM.CNT               VARIABLE REFERENCED AT :   273.000

ACI                     LABEL    REFERENCED AT :  7329.000

ACK.INTR                LABEL    REFERENCED AT :   817.000  819.000

ADD.SUB.FLT.PT          LABEL    REFERENCED AT :  6716.000 8151.000

ADDR.STOP               LABEL    REFERENCED AT :  1039.000 2637.000 2637.000

ADDRSTOP                STRING   REFERENCED AT :  1036.000 1084.000 2635.000

ADF                     LABEL    REFERENCED AT :  3580.000 8360.000

ADFD                    LABEL    REFERENCED AT :  3585.000 4022.000

ADFW                    LABEL    REFERENCED AT :  3582.000 3588.000

ADJ.B1                  LABEL    REFERENCED AT :  4228.000 4257.000 4299.000

ADJ.B2                  LABEL    REFERENCED AT :  4216.000 4242.000 4301.000 4303.000

ADVANCE.ON.BUS          LABEL    REFERENCED AT :  7381.000 7389.000

AEXP                    STRING   REFERENCED AT :   716.000 8433.000 8504.000 8505.000

AEXP.STATUS.IPII        LABEL    REFERENCED AT :  1742.000 1792.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
AI                          LABEL    REFERENCED AT :  2604.000 8573.000 8574.000

AI.EXIT                     LABEL    REFERENCED AT :  2611.000 2622.000

AI.INST                     LABEL    REFERENCED AT :  2227.000

AICT                        LABEL    REFERENCED AT :  7561.000

AICT                        STRING   REFERENCED AT :  1998.000 7387.000

AICT.DEACT                  LABEL    REFERENCED AT :  6395.000 6401.000

AICT.ICT                    LABEL    REFERENCED AT :  7339.000 7460.000 7548.000 7730.000

AICT.ICT.EXIT               LABEL    REFERENCED AT :  7613.000 7628.000 7639.000

AICT.RETRY                  LABEL    REFERENCED AT :  7562.000 7568.000

AL                          STRING   REFERENCED AT :  3107.000 3572.000 3574.000 3648.000 3785.000 3788.000 4492.000 4507.000
                                     4530.000 4708.000 4712.000 4715.000 4941.000 7074.000 8252.000 8292.000 8293.000 8294.000
                                     8299.000 8302.000 8303.000 8304.000 8305.000 8306.000 8324.000 8352.000 8353.000 8360.000
                                     8361.000 8378.000 8398.000 8399.000 8400.000 8424.000 8425.000 8426.000 8429.000 8432.000
                                     8434.000 8491.000 8492.000 8493.000 8496.000 8497.000 8500.000 8501.000

ALEVEL                      VARIABLE REFERENCED AT :   163.000  906.000 1192.000 1840.000 2520.000 2566.000 3166.000

ALTER.EXTENDFD.STATUS       LABEL    REFERENCED AT :  3389.000 4403.000 5052.000

ALU4-77                     STRING   REFERENCED AT :   814.000 1902.000 2180.000 2263.000 2462.000 2544.000 2729.000 2734.000
                                     2743.000 2744.000 2747.000 2748.000 3899.000 3926.000 4012.000 6035.000 6153.000 6308.000
                                     7234.000

ALUNEG                      STRING   REFERENCED AT :  1048.000 1444.000 1445.000 1448.000 1449.000 2144.000 2714.000 2765.000
                                     2928.000 3547.000 3552.000 3554.000 3557.000 3600.000 3771.000 3774.000 4030.000 4038.000
                                     4062.000 4080.000 4481.000 4487.000 4763.000 5247.000 7618.000 7818.000

ALUNEGW                     STRING   REFERENCED AT :  1063.000 1680.000 2294.000 2364.000 2563.000 2713.000 2905.000 2940.000
                                     2943.000 3599.000 5162.000 5313.000

ALUZ                        STRING   REFERENCED AT :   701.000  860.000  957.000 1004.000 1047.000 1279.000 1367.000 1368.000
                                     1370.000 1439.000 1440.000 1441.000 1461.000 1523.000 1552.000 1553.000 1651.000 1652.000
                                     1681.000 1754.000 1769.000 1785.000 1817.000 1818.000 1819.000 1897.000 2089.000 2145.000
                                     2239.000 2240.000 2241.000 2267.000 2270.000 2293.000 2300.000 2326.000 2329.000 2539.000
                                     2610.000 2622.000 2668.000 2692.000 2770.000 2773.000 2807.000 2827.000 2836.000 2840.000
                                     2862.000 3161.000 3532.000 3561.000 3797.000 3809.000 3853.000 3891.000 3898.000 3928.000
                                     4037.000 4142.000 4186.000 4199.000 4268.000 4286.000 4287.000 4561.000 4585.000 4586.000
                                     4596.000 4599.000 4604.000 4762.000 5136.000 5159.000 5252.000 5326.000 5434.000 5547.000
                                     5799.000 5802.000 5804.000 5822.000 5934.000 5956.000 5987.000 5996.000 6010.000 6041.000
                                     6053.000 6077.000 6092.000 6097.000 6162.000 6328.000 7426.000 7609.000 7619.000 7620.000
                                     7704.000 7836.000 7842.000 8034.000

ANM                         LABEL    REFERENCED AT :  4703.000 8292.000

ANM.BYTE                    LABEL    REFERENCED AT :  4711.000
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

ANM.DWORD	LABEL	REFERENCED AT :	4653.000 4704.000
ANM.HWORD	LABEL	REFERENCED AT :	4706.000
ANM.WORD	LABEL	REFERENCED AT :	4710.000 4714.000
ANM.WORD.BYTE	LABEL	REFERENCED AT :	705.000 4709.000
ARITHMETIC.EXCEPTION	LABEL	REFERENCED AT :	717.000 1733.000
ARM	LABEL	REFERENCED AT :	4935.000 8324.000
ARM.DWORD	LABEL	REFERENCED AT :	4937.000
ARM.FETCH.MSW	LABEL	REFERENCED AT :	4820.000 4938.000
ARM.HWORD	LABEL	REFERENCED AT :	4951.000
ARM.W.H.B	LABEL	REFERENCED AT :	4936.000 4947.000
ARM.WORD.BYTE	LABEL	REFERENCED AT :	4949.000
ARSTX	LABEL	REFERENCED AT :	7489.000
ARSTX	STRING	REFERENCED AT :	1971.000 7381.000
ARSTX.RETRY	LABEL	REFERENCED AT :	7490.000 7495.000
ARSTX.RSTX	LABEL	REFERENCED AT :	7347.000 7476.000 7695.000 7739.000
ARSTX.RSTX.JUMP	LABEL	REFERENCED AT :	7313.000 7325.000 7328.000 7331.000
ASFPW	STRING	REFERENCED AT :	6771.000 6779.000 6788.000 6796.000 6832.000 6840.000 6849.000 6857.000 6897.000 6907.000 6916.000 6926.000 6954.000 6964.000 6974.000 6984.000 7011.000 7019.000 7078.000 7081.000 7084.000 7117.000 7127.000 7175.000 7180.000 7184.000
B	STRING	REFERENCED AT :	8302.000 8319.000 8321.000 8398.000 8421.000 8429.000
B.IOCD1	VARIABLE	REFERENCED AT :	279.000 1299.000 1375.000
B.IOCD2	VARIABLE	REFERENCED AT :	212.000 1302.000 1374.000
BACKDATE.PC	LABEL	REFERENCED AT :	660.000 1040.000 1131.000 5254.000
BADSCALE	STRING	REFERENCED AT :	3644.000
BCF	LABEL	REFERENCED AT :	5061.000 8155.000
BCT	LABEL	REFERENCED AT :	5088.000 8154.000
BEI	LABEL	REFERENCED AT :	4396.000 8381.000
BEI.1	LABEL	REFERENCED AT :	4398.000 6452.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

BEI.TEST	LABEL	REFERENCED AT : 4397.000 6448.000
BFT	LABEL	REFERENCED AT : 5154.000 8514.000
BFT2	LABEL	REFERENCED AT : 5159.000 5162.000
BIB	LABEL	REFERENCED AT : 5126.000 8615.000
BIBUSY	STRING	REFERENCED AT : 871.000 1134.000 1891.000 1892.000 1950.000 1975.000 2140.000 2233.000 3730.000 3744.000 3757.000 4568.000 5898.000 6664.000
BID	LABEL	REFERENCED AT : 5129.000 8627.000
BIH	LABEL	REFERENCED AT : 5127.000 8619.000
BINC	LABEL	REFERENCED AT : 5124.000 8156.000
BINC2	LABEL	REFERENCED AT : 5131.000 5135.000
BINC3	LABEL	REFERENCED AT : 5130.000 5136.000
BINC4	LABEL	REFERENCED AT : 5136.000 5144.000
BIT	STRING	REFERENCED AT : 4539.000 4544.000 4977.000 5004.000 8250.000 8251.000 8253.000 8297.000 8298.000 8300.000
BIT.MEM	LABEL	REFERENCED AT : 4961.000 8132.000 8133.000 8134.000
BIT.MEM.EXIT	LABEL	REFERENCED AT : 4976.000
BIT.MEM.INDIRECT	LABEL	REFERENCED AT : 4965.000 4971.000
BIT.TBM	LABEL	REFERENCED AT : 4990.000 8135.000
BIT.TBM.INDIRECT	LABEL	REFERENCED AT : 4994.000 5000.000
BITCNT	STRING	REFERENCED AT : 7055.000 7057.000 7147.000 7149.000
BITREGS	LABEL	REFERENCED AT : 4534.000 8098.000 8099.000 8100.000 8101.000 8176.000 8177.000 8178.000 8179.000
BIW	LABEL	REFERENCED AT : 5128.000 8623.000
BL	LABEL	REFERENCED AT : 5111.000 8544.000
BL2	LABEL	REFERENCED AT : 5113.000
BL3	LABEL	REFERENCED AT : 5112.000 5115.000
BL4	LABEL	REFERENCED AT : 5115.000 5117.000
BLK.ERR.EXIT	LABEL	REFERENCED AT : 3135.000 3142.000
BLK.MODE.ERR	LABEL	REFERENCED AT : 3136.000 5418.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

BLKCAR8	STRING	REFERENCED AT : 3541.000 3543.000 3545.000 3546.000 3550.000 3551.000 3584.000 3598.000 3640.000 3848.000 3850.000 3852.000 3894.000 3895.000 3922.000 3923.000 4027.000 4029.000 4036.000 4060.000 4061.000 4201.000 4202.000
BLOCK.MODE.TIMEOUT	LABEL	REFERENCED AT : 723.000 3138.000
BMG	REGISTER	REFERENCED AT : 4534.000 4539.000 4544.000 4545.000
BMGMR	STRING	REFERENCED AT : 4964.000 4970.000 4977.000 4993.000 4999.000
BMUX	STRING	REFERENCED AT : 94.000 906.000 1930.000 2045.000 2719.000 4063.000 5525.000 5759.000 6390.000 6993.000 7049.000 7139.000
BMUX00	STRING	REFERENCED AT : 926.000 1017.000 1044.000 1349.000 1757.000 1808.000 2066.000 2181.000 2256.000 3779.000 3781.000 5749.000 5760.000 5886.000 5926.000 6038.000 6039.000 6045.000 6121.000 6128.000 6210.000 6317.000 6471.000 7477.000 7549.000
BMUX16	STRING	REFERENCED AT : 866.000 964.000 1089.000 1324.000 1377.000 1401.000 1611.000 1756.000 1768.000 2411.000 2454.000 2719.000 2755.000 2776.000 2815.000 2850.000 2880.000 2946.000 5345.000 5814.000 6102.000 6383.000 7453.000 7664.000 7953.000
BMUX17	STRING	REFERENCED AT : 948.000 966.000 2193.000 2338.000 2720.000 2756.000 2881.000 5347.000 5815.000 6385.000 6386.000 6425.000 7454.000 8016.000
BMUX18	STRING	REFERENCED AT : 693.000 996.000 2163.000 2168.000 2272.000 2735.000 2816.000 2851.000 5257.000 5816.000 7357.000 7599.000 7613.000
BMUX19	STRING	REFERENCED AT : 817.000 907.000 933.000 2028.000 2167.000 2271.000 2296.000 5354.000 5817.000 7341.000 8010.000
BORROW.SEGMENTS	LABEL	REFERENCED AT : 6471.000 6478.000
BR1	LABEL	REFERENCED AT : 5090.000 5162.000 8515.000 8516.000 8517.000 8518.000 8519.000 8520.000 8521.000
BR1.ERR	LABEL	REFERENCED AT : 5078.000 5091.000 5124.000 5185.000 5233.000
BR2	LABEL	REFERENCED AT : 5093.000
BR3	LABEL	REFERENCED AT : 2567.000 5081.000 5092.000 5094.000 5116.000 5326.000
BRANCH	STRING	REFERENCED AT : 4527.000 4551.000
BRANCH.INDIRECT	LABEL	REFERENCED AT : 5093.000 5113.000 5204.000 5216.000
BRANCH.INDIRECT.EXIT	LABEL	REFERENCED AT : 5212.000 5218.000
BRANCH.NOT.TAKEN	LABEL	REFERENCED AT : 5067.000 5089.000
BRI	LABEL	REFERENCED AT : 2511.000 8545.000
BRI.55.HALT	LABEL	REFERENCED AT : 3345.000
BRI.75.EXIT	LABEL	REFERENCED AT : 1843.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

BRI.DACI	LABEL	REFERENCED AT :	7272.000 7336.000
BRI.DAT	LABEL	REFERENCED AT :	2542.000
BRI.ERR.FLG	VARIABLE	REFERENCED AT :	548.000 3354.000 7274.000
BRI.ERROR	LABEL	REFERENCED AT :	2548.000 3352.000
BRI.EXIT	LABEL	REFERENCED AT :	2517.000 2531.000
BRI.FORMAT.PSW	LABEL	REFERENCED AT :	1834.000 2564.000
BRI.INDR	VARIABLE	REFERENCED AT :	243.000 2527.000 2532.000 2550.000
BRI.JUMP	LABEL	REFERENCED AT :	2539.000 2547.000 2549.000 7290.000
BRI.LDPC	LABEL	REFERENCED AT :	2552.000 2565.000
BRI.RTN	LABEL	REFERENCED AT :	2514.000 6742.000
BRI0	LABEL	REFERENCED AT :	2526.000 2528.000 2533.000
BRI1	LABEL	REFERENCED AT :	2527.000 2534.000
BUILD.INTR.ADDR	LABEL	REFERENCED AT :	803.000 1747.000 2536.000 2574.000 2585.000 2594.000 2606.000 2619.000 2659.000 5241.000
BUILD.INTR.EXIT	LABEL	REFERENCED AT :	2680.000 2690.000 2693.000
BUILD.IOCD	LABEL	REFERENCED AT :	2317.000 2331.000 2350.000
BUILD.VECTOR	LABEL	REFERENCED AT :	1255.000 1262.000
BUMP	VARIABLE	REFERENCED AT :	169.000 6466.000 6543.000 6586.000 6594.000 6625.000 6642.000
BUSREQ	VARIABLE	REFERENCED AT :	250.000 1310.000 2066.000 2070.000 2181.000 2184.000 2256.000 7414.000 7477.000 7484.000 7536.000 7549.000 7556.000 7601.000
BUSY.FLG	VARIABLE	REFERENCED AT :	531.000 594.000 3006.000 3027.000 7525.000 7569.000
BYTE	STRING	REFERENCED AT :	1951.000 2045.000 2472.000 4710.000 5457.000 8054.000
C	STRING	REFERENCED AT :	4555.000 4727.000 4731.000 4733.000 8295.000 8494.000
CALM	LABEL	REFERENCED AT :	5231.000 8106.000 8184.000
CALM.EXIT	LABEL	REFERENCED AT :	2165.000 2193.000
CALM.EXTL	LABEL	REFERENCED AT :	2028.000 2028.000 2068.000 2068.000 2160.000
CALM.INT.TEST	LABEL	REFERENCED AT :	4433.000 5234.000
CALM.RET	LABEL	REFERENCED AT :	4438.000 5236.000
CALM.TRAP	LABEL	REFERENCED AT :	3168.000 5255.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

CALM.TST	LABEL	REFERENCED AT :	860.000 995.000
CAM	LABEL	REFERENCED AT :	4722.000 8295.000
CAM.B.H.W	LABEL	REFERENCED AT :	4723.000 4728.000
CAM.DWORD	LABEL	REFERENCED AT :	4724.000
CAM.FETCH.MSW	LABEL	REFERENCED AT :	4725.000 4821.000
CAM.HWORD	LABEL	REFERENCED AT :	4730.000
CAM.WORD.BYTE	LABEL	REFERENCED AT :	4729.000 4732.000
CAR	LABEL	REFERENCED AT :	4520.000 8494.000
CATASTROPHIC.ERROR	LABEL	REFERENCED AT :	7667.000 7673.000
CC3	VARIABLE	REFERENCED AT :	559.000 7502.000 7574.000 7677.000
CC4	VARIABLE	REFERENCED AT :	560.000 7496.000
CCTEST	STRING	REFERENCED AT :	5061.000 5089.000
CD	LABEL	REFERENCED AT :	2259.000 8579.000 8580.000
CD.2ND.TERMINATE	LABEL	REFERENCED AT :	2428.000 2431.000
CD.CMD	LABEL	REFERENCED AT :	2269.000 2287.000
CD.CNTRL	LABEL	REFERENCED AT :	2270.000 2288.000
CD.DUD	LABEL	REFERENCED AT :	1976.000 2009.000 2025.000 2092.000 2127.000 2142.000 2152.000 2179.000 2206.000 2217.000 2235.000 2256.000 8052.000
CD.ERROR.EXIT	LABEL	REFERENCED AT :	2409.000 2414.000
CD.EXIT	LABEL	REFERENCED AT :	1805.000 2180.000 2276.000 2415.000 2433.000 2436.000 2451.000 2459.000 2962.000
CD.NORMAL.EXIT	LABEL	REFERENCED AT :	2408.000 2412.000 2478.000
CD.TCWA	LABEL	REFERENCED AT :	2272.000 2445.000 2454.000
CD.TCWA.IO	LABEL	REFERENCED AT :	2447.000 2452.000 2456.000
CD.TERMINATE	LABEL	REFERENCED AT :	2271.000 2421.000
CDECRN	STRING	REFERENCED AT :	1110.000 1118.000 4914.000 6543.000
CEA	LABEL	REFERENCED AT :	5049.000 8390.000
CHAN.WCS.N.FLG	VARIABLE	REFERENCED AT :	543.000 7434.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

CHANGE.EXTENDED.MODE	LABEL	REFERENCED AT : 5048.000 5051.000
CHANGE.TO.DOUBLE	LABEL	REFERENCED AT : 6755.000 6816.000 6994.000 7025.000 7050.000
CHANGE.TO.SINGLE	LABEL	REFERENCED AT : 6880.000 6937.000 7035.000 7095.000 7140.000
CHANNEL.NOT.PRESENT	LABEL	REFERENCED AT : 7664.000 7676.000
CHECK.CHANNEL.IPL.STATS	LABEL	REFERENCED AT : 7741.000
CHECK.CLASSF.IPL	LABEL	REFERENCED AT : 1323.000 7681.000
CHECK.CPU.MODE	LABEL	REFERENCED AT : 2490.000 4388.000 4408.000 4424.000 5294.000 5451.000 6451.000 6653.000
CHECK.DISABLE.AEXP	LABEL	REFERENCED AT : 1734.000 1736.000
CHECK.DRT	LABEL	REFERENCED AT : 2232.000 2243.000 2247.000
CHECK.MAP.CLEAR	LABEL	REFERENCED AT : 6476.000 6532.000
CHECK.NOT.55.ENBL	LABEL	REFERENCED AT : 4423.000 7977.000 8007.000
CHECK.STOP	LABEL	REFERENCED AT : 1034.000
CI	LABEL	REFERENCED AT : 4554.000 8403.000
CK.BLK.TIMEOUT	LABEL	REFERENCED AT : 721.000
CK.BUSY1	LABEL	REFERENCED AT : 3002.000 3004.000
CK.IOTIMEOUT	LABEL	REFERENCED AT : 2999.000 3001.000
CK.OPRNDPE	LABEL	REFERENCED AT : 3056.000 3058.000
CK.RETRY1	LABEL	REFERENCED AT : 3005.000 3007.000
CK.TIMEOUT	LABEL	REFERENCED AT : 3053.000 3055.000
CLASS.012	LABEL	REFERENCED AT : 2463.000
CLASS.012DF.IPL.IOCD	LABEL	REFERENCED AT : 1296.000
CLASSF.INT.ERROR	LABEL	REFERENCED AT : 7461.000 7469.000
CLASSF.INT.RTN	LABEL	REFERENCED AT : 861.000
CLDNU	STRING	REFERENCED AT : 1084.000 1085.000 1221.000 1233.000 1899.000 2077.000 2222.000 2550.000 2663.000 2710.000 2766.000 2771.000 2774.000 2777.000 2837.000 2841.000 2937.000 2940.000 2943.000 2944.000 2946.000 3211.000 3853.000 3928.000 3987.000 4285.000 4766.000 5111.000 5246.000 5457.000 5464.000 5827.000 6096.000 6114.000 6149.000 6481.000 6505.000 6609.000 6654.000 7423.000
CLEAR.GRANULARITY	LABEL	REFERENCED AT : 5355.000 5359.000
CLEAR.MEM.ERROR	LABEL	REFERENCED AT : 1949.000 2051.000 5471.000 5911.000 8060.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U


```
CLEAR.TIMEOUT                   LABEL    REFERENCED AT :   2188.000 2278.000 2978.000 3357.000 7951.000

CLKDIV                          STRING   REFERENCED AT :   3963.000

CLR.SCRFLGS                     LABEL    REFERENCED AT :   1197.000 1206.000 1209.000 1218.000 1220.000 1222.000 1226.000 1229.000
                                         1253.000 1259.000 1466.000 1475.000

CLRS                            STRING   REFERENCED AT :    597.000  652.000  682.000  767.000  770.000 1102.000 1151.000 1193.000
                                         1230.000 1300.000 1424.000 1433.000 2192.000 2329.000 2742.000 2763.000 2767.000 2927.000
                                         2939.000 3124.000 3146.000 3411.000 3502.000 3762.000 3764.000 3805.000 3894.000 3919.000
                                         3929.000 4025.000 4221.000 4250.000 4280.000 4298.000 4308.000 4471.000 4616.000 5050.000
                                         6009.000 6638.000 7984.000

CLRS.DUD                        LABEL    REFERENCED AT :   4222.000 4251.000 4279.000

CLRS.OB.DUD                     LABEL    REFERENCED AT :   4208.000 4210.000 4307.000

CLRSYSR                         STRING   REFERENCED AT :   1124.000 1172.000

CLRTO                           STRING   REFERENCED AT :   1317.000 1628.000 1703.000 2080.000 2981.000 3103.000 3122.000 5911.000
                                         5916.000 7494.000

CMD                             VARIABLE REFERENCED AT :    211.000 1811.000 1814.000 2266.000 2299.000 2341.000 2371.000 2394.000

CMD.AICT                        LABEL    REFERENCED AT :   1997.000 2204.000 6398.000

CMD.ARSTX                       LABEL    REFERENCED AT :    825.000 1970.000 2125.000

CMD.ICT                         LABEL    REFERENCED AT :   2004.000 2215.000 6416.000

CMD.IOC                         LABEL    REFERENCED AT :   2064.000 2402.000 2453.000 2730.000 2738.000

CMD.RSTX                        LABEL    REFERENCED AT :    845.000 1800.000 1983.000 2136.000

CMD.WDOT                        LABEL    REFERENCED AT :   1620.000 1620.000 1706.000 1990.000 2147.000 2432.000

CMM                             LABEL    REFERENCED AT :   4740.000 8296.000

CMM.B.H.W                       LABEL    REFERENCED AT :   4741.000 4749.000

CMM.DWORD                       LABEL    REFERENCED AT :   4742.000

CMM.EXIT                        LABEL    REFERENCED AT :   4747.000 4752.000

CMM.FETCH.MSW                   LABEL    REFERENCED AT :   4743.000 4829.000

CMM.HWORD                       LABEL    REFERENCED AT :   4751.000

CMM.WORD.BYTE                   LABEL    REFERENCED AT :   4745.000 4750.000

CMR                             LABEL    REFERENCED AT :   4516.000 8495.000

CNT.FXH.FLG                     VARIABLE REFERENCED AT :    535.000  594.000 3006.000 3008.000 7519.000 7569.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

COMPLAF	STRING	REFERENCED AT : 7003.000 7067.000 7159.000
COMPLAN	STRING	REFERENCED AT : 6991.000 6996.000 7047.000 7052.000 7137.000 7144.000
COMPLAP	STRING	REFERENCED AT : 7108.000
COMPUTE.CPU.STATUS	LABEL	REFERENCED AT : 3244.000 7957.000 7979.000 8028.000
CONNECT.ERROR.TRAP	LABEL	REFERENCED AT : 7865.000 7870.000
CONSOLE.INTERRUPT	LABEL	REFERENCED AT : 747.000 773.000 1729.000
CORRMNG	STRING	REFERENCED AT : 6778.000 6795.000 6839.000 6856.000 6906.000 6925.000 6963.000 6983.000 7018.000 7077.000 7126.000 7174.000
CORRZEROW	STRING	REFERENCED AT : 6991.000 6996.000 7047.000 7052.000 7092.000 7099.000 7137.000 7144.000
COUNT.DOWN	LABEL	REFERENCED AT : 822.000 2075.000 2078.000 2080.000 2124.000 2201.000 7487.000 7559.000
CPAOB	STRING	REFERENCED AT : 6752.000 6813.000 6875.000 6932.000 6989.000 7045.000 7059.000 7089.000 7133.000 7151.000
CPSOB	STRING	REFERENCED AT : 6876.000 6933.000 7090.000 7134.000
CPSTS	VARIABLE	REFERENCED AT : 213.000 800.000 866.000 997.000 997.000 3224.000 3309.000 3388.000 6442.000
CR.TD4	LABEL	REFERENCED AT : 2819.000
CR.TD8	LABEL	REFERENCED AT : 2815.000 2832.000
CROM.ADDR	LABEL	REFERENCED AT : 5486.000
CTL	LABEL	REFERENCED AT : 4464.000 8090.000 8168.000
CURRENT.I.EXIT	LABEL	REFERENCED AT : 3054.000 3057.000 3060.000 3064.000
CURRENT.INST.ERROR	LABEL	REFERENCED AT : 946.000 1911.000 1940.000 2682.000 3052.000 3229.000 3421.000 3476.000 3522.000 4637.000 4789.000 4818.000 4827.000 4835.000 4849.000 4880.000 4897.000 4922.000 4944.000 4969.000 4998.000 5194.000 5211.000 5311.000 5513.000 5513.000 6498.000 6568.000 6586.000
CURRENT.PSW	LABEL	REFERENCED AT : 7453.000 7465.000
D	STRING	REFERENCED AT : 3503.000 3527.000 3826.000 4140.000 4744.000 4816.000 4825.000 6995.000 6999.000 7001.000 7002.000 7098.000 7102.000 7104.000 7105.000
D.ADDR	VARIABLE	REFERENCED AT : 207.000 2043.000 2369.000 2467.000
DACI	LABEL	REFERENCED AT : 7332.000
DACI.EXIT	LABEL	REFERENCED AT : 7344.000 7356.000
DAE	LABEL	REFERENCED AT : 3496.000 8389.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

DAI	LABEL	REFERENCED AT : 2617.000 8575.000 8576.000
DAI.INSTR	LABEL	REFERENCED AT : 2226.000 2236.000
DAI.INTFF.FLAG	LABEL	REFERENCED AT : 2624.000 2638.000
DCI	LABEL	REFERENCED AT : 7326.000
DEACT.WAIT.READY	LABEL	REFERENCED AT : 6399.000 6404.000
DECODE	LABEL	REFERENCED AT : 638.000
DECODE(	STRING	REFERENCED AT : 639.000 682.000 683.000 1091.000 1092.000 1096.000 1097.000 4500.000 4501.000 4525.000 4526.000 4540.000 4541.000 4545.000 4546.000 4573.000 4574.000 5098.000 5099.000 5141.000 5142.000 5175.000 5176.000 5177.000 5178.000 5198.000 5199.000 5201.000 5202.000 5296.000 5329.000 5330.000 6719.000 6727.000 6733.000 6753.000 6814.000 6876.000 6933.000 6990.000 7026.000 7036.000 7046.000 7091.000 7133.000 7169.000 7954.000 7955.000
DECREMENT.PC	LABEL	REFERENCED AT : 3068.000 3137.000 3176.000 3202.000 3265.000 3353.000 3416.000 7766.000 7855.000 7909.000
DECRN	STRING	REFERENCED AT : 826.000 840.000 849.000 1154.000 1186.000 1259.000 1305.000 1392.000 1395.000 2079.000 2980.000 3587.000 3651.000 3687.000 3691.000 3696.000 3708.000 3836.000 3963.000 3967.000 3969.000 4024.000 4041.000 4097.000 4107.000 4111.000 4118.000 4131.000 4145.000 4148.000 4149.000 4150.000 4151.000 4152.000 4289.000 4561.000 4569.000 4611.000 4782.000 5158.000 5159.000 5160.000 5618.000 5856.000 5947.000 6399.000 6408.000 6420.000 6516.000 6590.000 6599.000 6643.000 7001.000 7065.000 7104.000 7157.000 7495.000 7508.000 7518.000 7568.000 7581.000 7593.000 7661.000
DEF.75H	FILE	REFERENCED AT : 632.000
DELAY1	LABEL	REFERENCED AT : 3536.000 4504.000 4587.000 5131.000
DELAYED.BI.BUSY.TEST	LABEL	REFERENCED AT : 1964.000 1972.000 1991.000 1998.000 2015.000
DELSHF	STRING	REFERENCED AT : 6756.000 6817.000 6882.000 6939.000
DELSHFL	STRING	REFERENCED AT : 6756.000 6817.000 6882.000 6939.000
DEV.CLASS.012	LABEL	REFERENCED AT : 2295.000
DEV.CLASS0	LABEL	REFERENCED AT : 2307.000
DEV.CLASS1	LABEL	REFERENCED AT : 2300.000 2323.000
DEV.CLASS1.EXIT	LABEL	REFERENCED AT : 2326.000 2330.000 2341.000 2344.000
DEV.CLASS2	LABEL	REFERENCED AT : 2301.000 2337.000
DEV.CLASS3	LABEL	REFERENCED AT : 1806.000 2267.000
DEV.CLASSE	LABEL	REFERENCED AT : 2293.000 2393.000
DEV.CLASSF.INT	LABEL	REFERENCED AT : 815.000 7450.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

DEXP.GT.13.ADD.DOUBLE	LABEL	REFERENCED AT : 6887.000
DEXP.GT.SIX.ADD.SING	LABEL	REFERENCED AT : 6761.000
DEXP.GT.SIX.SUB.DOUBLE	LABEL	REFERENCED AT : 6944.000
DEXP.GT.SIX.SUB.SING	LABEL	REFERENCED AT : 6822.000
DEXP.LT.13.ADD.DOUBLE	LABEL	REFERENCED AT : 6910.000
DEXP.LT.SIX.ADD.SING	LABEL	REFERENCED AT : 6781.000
DEXP.LT.SIX.SUB.DOUBLE	LABEL	REFERENCED AT : 6967.000
DEXP.LT.SIX.SUB.SING	LABEL	REFERENCED AT : 6842.000
DEXP.ZERO.ADD.DOUBLE	LABEL	REFERENCED AT : 6891.000
DEXP.ZERO.ADD.SING	LABEL	REFERENCED AT : 6765.000
DEXP.ZERO.SUB.DOUBLE	LABEL	REFERENCED AT : 6948.000
DEXP.ZERO.SUB.SING	LABEL	REFERENCED AT : 6826.000
DI	STRING	REFERENCED AT : 957.000 1105.000 1111.000 1216.000 1221.000 1223.000 1227.000 1264.000 1285.000 1397.000 1435.000 1453.000 1461.000 1467.000 1572.000 1737.000 1895.000 1951.000 1952.000 2100.000 2193.000 2311.000 2316.000 2467.000 2776.000 2778.000 2782.000 2886.000 2890.000 2905.000 2906.000 2957.000 3171.000 3307.000 3502.000 3603.000 3604.000 3670.000 3731.000 3738.000 3745.000 3752.000 3758.000 3766.000 3778.000 3805.000 3832.000 3851.000 3658.000 3904.000 3913.000 3919.000 3924.000 3927.000 3936.000 3942.000 3961.000 3964.000 3991.000 3999.000 4001.000 4009.000 4076.000 4078.000 4085.000 4120.000 4153.000 4185.000 4190.000 4197.000 4216.000 4228.000 4233.000 4242.000 4246.000 4257.000 4260.000 4263.000 4569.000 4918.000 4923.000 5598.000 5602.000 5606.000 5792.000 5793.000 5835.000 5851.000 5867.000 5877.000 5931.000 5939.000 6025.000 6049.000 6091.000 6140.000 6158.000 6183.000 6248.000 6311.000 6483.000 6541.000 6667.000 7239.000 7256.000 7270.000 7294.000 7297.000 7300.000 7303.000 7306.000 7309.000 7312.000 7325.000 7328.000 7331.000 7337.000 7430.000 7456.000 7459.000 7601.000 7694.000 7729.000 7738.000 7751.000
DI	REGISTER	REFERENCED AT : 94.000 936.000 951.000 952.000 953.000 963.000 964.000 965.000 966.000 969.000 1111.000 1119.000 1125.000 1223.000 1227.000 1263.000 1264.000 1276.000 1316.000 1322.000 1323.000 1344.000 1345.000 1355.000 1358.000 1361.000 1363.000 1398.000 1433.000 1437.000 1438.000 1439.000 1442.000 1443.000 1446.000 1447.000 1450.000 1453.000 1454.000 1517.000 1575.000 1656.000 1665.000 1675.000 1677.000 1678.000 1681.000 1686.000 1723.000 1739.000 1767.000 1772.000 1819.000 1842.000 1900.000 1904.000 1908.000 1930.000 1937.000 1952.000 1953.000 2045.000 2058.000 2141.000 2144.000 2145.000 2166.000 2167.000 2309.000 2315.000 2408.000 2411.000 2448.000 2461.000 2462.000 2472.000 2495.000 2529.000 2531.000 2535.000 2688.000 2693.000 2719.000 2733.000 2740.000 2890.000 2898.000 2906.000 2917.000 2955.000 2958.000 3174.000 3225.000 3226.000 3231.000 3308.000 3417.000 3519.000 3536.000 3537.000 3543.000 3574.000 3597.000 3603.000 3604.000 3610.000 3640.000 3660.000 3675.000 3737.000 3738.000 3751.000 3752.000 3762.000 3763.000 3778.000 3783.000 3784.000 3788.000 3804.000 3821.000 3822.000 3837.000 3848.000 3849.000 3851.000 3855.000 3878.000 3893.000 3896.000 3898.000 3906.000 3910.000 3912.000 3918.000 3924.000 3931.000 3932.000 3942.000 3961.000 3964.000 3992.000 3993.000 3996.000 4000.000 4003.000 4004.000 4008.000 4009.000 4010.000 4026.000 4050.000 4063.000 4065.000 4078.000 4085.000 4103.000 4184.000 4193.000 4196.000 4240.000 4267.000 4270.000 4271.000 4273.000 4301.000 4302.000 4305.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

		4476.000 4571.000 4634.000 4657.000 4661.000 4674.000 4687.000 4708.000 4712.000 4715.000 4727.000 4731.000 4733.000 4746.000 4752.000 4782.000 4796.000 4814.000 4823.000 4831.000 4846.000 4873.000 4911.000 4923.000 4941.000 4950.000 4952.000 4966.000 4975.000 4977.000 4981.000 4982.000 4995.000 5004.000 5131.000 5132.000 5191.000 5205.000 5206.000 5222.000 5223.000 5299.000 5307.000 5317.000 5456.000 5458.000 5525.000 5531.000 5533.000 5570.000 5606.000 5617.000 5793.000 5794.000 5842.000 5844.000 5856.000 5887.000 5903.000 5915.000 5927.000 5934.000 5935.000 5940.000 6029.000 6031.000 6040.000 6046.000 6054.000 6097.000 6122.000 6129.000 6141.000 6165.000 6177.000 6184.000 6211.000 6249.000 6318.000 6390.000 6468.000 6471.000 6481.000 6485.000 6500.000 6505.000 6546.000 6595.000 6668.000 6675.000 6727.000 6728.000 6730.000 6731.000 6880.000 6937.000 6993.000 7026.000 7036.000 7049.000 7097.000 7139.000 7142.000 7256.000 7294.000 7297.000 7300.000 7303.000 7306.000 7309.000 7312.000 7325.000 7328.000 7430.000 7477.000 7529.000 7530.000 7549.000 7599.000 7600.000 7602.000 7652.000 7672.000 7682.000 7684.000 7685.000 7701.000 7715.000 7746.000 7752.000 8049.000 8054.000
DI.INST	LABEL	REFERENCED AT : 2583.000 8569.000 8570.000
DINTRA	STRING	REFERENCED AT : 916.000 2229.000 2241.000 3323.000 7620.000 7624.000 7640.000
DIS.BLK.TIMEOUT	STRING	REFERENCED AT : 1170.000 8024.000 8038.000
DISPATCH.AICT.ICT	LABEL	REFERENCED AT : 7338.000
DISPATCH.ARSTX.RSTX	LABEL	REFERENCED AT : 7295.000 7298.000 7301.000 7307.000 7310.000 7314.000 7346.000 7437.000
DISPATCH.CLEAR	LABEL	REFERENCED AT : 1468.000 1470.000 1474.000
DISPATCH.EXT.DEACTIVATE	LABEL	REFERENCED AT : 2545.000 7269.000
DISPATCH.WDOT	LABEL	REFERENCED AT : 7304.000 7649.000
DIVIDE	STRING	REFERENCED AT : 3772.000 3876.000
DIVLOOP	STRING	REFERENCED AT : 7065.000 7157.000
DIVLSW	STRING	REFERENCED AT : 3961.000 3964.000
DIVMSW	STRING	REFERENCED AT : 3962.000 3965.000
DIX	STRING	REFERENCED AT : 1575.000 2531.000 3519.000 3537.000 4634.000 4846.000 4873.000 4966.000 4995.000 5132.000 5191.000 5205.000 5222.000 5223.000 5299.000 5456.000 5458.000 5570.000 6728.000 8049.000
DO.INTR.CTL	LABEL	REFERENCED AT : 2178.000 2578.000 2599.000 2610.000 2625.000
DO.INTR.CTL.BUSY.LOOP	LABEL	REFERENCED AT : 2182.000 2186.000
DOUBLE.DUD	LABEL	REFERENCED AT : 2008.000
DP.DIV	LABEL	REFERENCED AT : 3943.000 3960.000 3977.000
DP.DIV.DONE	LABEL	REFERENCED AT : 3967.000 3989.000
DP.DIV.REENTRY	LABEL	REFERENCED AT : 3974.000 3981.000 3985.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S F L 3 2 / 7 5 C P U

DP.DIV.SUB	LABEL	REFERENCED AT : 3972.000 3980.000 3984.000 3986.000 3991.000
DP.Q.1	LABEL	REFERENCED AT : 3969.000 3983.000
DP.Q.2	LABEL	REFERENCED AT : 3970.000 3979.000
DPEFF	STRING	REFERENCED AT : 2561.000 3101.000 5317.000
DRDEF	FILE	REFERENCED AT : 8082.000
DSFPW	STRING	REFERENCED AT : 6753.000 6814.000 6877.000 6934.000 6990.000 7046.000 7091.000 7135.000
DUD.0EX	LABEL	REFERENCED AT : 1072.000 1074.000 1077.000
DUD.0XX	LABEL	REFERENCED AT : 684.000 1059.000 1759.000
DUD.1XX	LABEL	REFERENCED AT : 1260.000 1387.000 1458.000 1459.000
DUD.2	LABEL	REFERENCED AT : 1593.000 1664.000 1709.000 1725.000 1739.000 1908.000 1937.000
DUD.32X	LABEL	REFERENCED AT : 2056.000
DUD.50X	LABEL	REFERENCED AT : 2948.000 3350.000
DUD.5XX	LABEL	REFERENCED AT : 3966.000 3968.000 4029.000 4078.000 4112.000 4229.000 4231.000
DUD.62X	LABEL	REFERENCED AT : 3560.000 3575.000 3802.000 3910.000
DUD.65X	LABEL	REFERENCED AT : 3646.000 3662.000 3720.000
DUD.A8X	LABEL	REFERENCED AT : 5798.000 5801.000 5803.000 5806.000
DUD.B6X	LABEL	REFERENCED AT : 5761.000 5958.000 5977.000 6040.000 6084.000 6129.000
DUD.CALM	LABEL	REFERENCED AT : 997.000 999.000
DUD.EDX	LABEL	REFERENCED AT : 7606.000
DUD.INTR	LABEL	REFERENCED AT : 2470.000 2530.000 2686.000 2691.000 2700.000 2740.000 2772.000
DUD.WAIT.DRT	LABEL	REFERENCED AT : 1955.000 2057.000 2144.000 5455.000
DVF	LABEL	REFERENCED AT : 3840.000 8353.000
DVFD	LABEL	REFERENCED AT : 3841.000 3909.000
DVFW	LABEL	REFERENCED AT : 3842.000
DVI	LABEL	REFERENCED AT : 3756.000 8402.000
DVM	LABEL	REFERENCED AT : 3749.000 8315.000
DVR	LABEL	REFERENCED AT : 3761.000 8268.000
DWAIT	STRING	REFERENCED AT : 5417.000 5440.000 6807.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

DWORD	STRING	REFERENCED AT : 4649.000 4670.000 4682.000 4695.000 4704.000 4723.000 4741.000 4873.000 4892.000 4936.000 5223.000
E	STRING	REFERENCED AT : 4748.000 4761.000 8296.000 8495.000
E.TD2	LABEL	REFERENCED AT : 2885.000
E.TD2.S	LABEL	REFERENCED AT : 2889.000
E.TD4	LABEL	REFERENCED AT : 2881.000 2896.000
E.TD4.S	LABEL	REFERENCED AT : 2899.000 2918.000
E.TD8	LABEL	REFERENCED AT : 2880.000 2913.000
EAE	LABEL	REFERENCED AT : 3486.000 8383.000
ECI	LABEL	REFERENCED AT : 7323.000
ECWCS	LABEL	REFERENCED AT : 7315.000
EDEV	VARIABLE	REFERENCED AT : 253.000 1890.000 2263.000 2275.000 2446.000 2667.000 2684.000 7243.000
EFF.ERR.NPM	LABEL	REFERENCED AT : 1582.000 1584.000
EFFECTIVE.ADDR.EXIT	LABEL	REFERENCED AT : 1579.000
EFFECTIVE.ADDR.INDIR	LABEL	REFERENCED AT : 1575.000 3474.000 8044.000
EI	LABEL	REFERENCED AT : 2572.000 8567.000 8568.000
EI.EXIT	LABEL	REFERENCED AT : 2577.000 2587.000
ELSA	STRING	REFERENCED AT : 7961.000 8008.000 8017.000
ENAINTFF	STRING	REFERENCED AT : 681.000 786.000 813.000 1081.000 1133.000 2640.000 3163.000 3167.000 7334.000 7697.000
ENATBMT	STRING	REFERENCED AT : 1166.000 1778.000 5823.000 5846.000
ENAUORD	STRING	REFERENCED AT : 692.000 982.000 983.000 1003.000 1029.000 1031.000 1057.000 1101.000 1104.000 1126.000 1127.000 1171.000 1181.000 1271.000 1274.000 1535.000 1724.000 1726.000 1856.000 1953.000 1954.000 3066.000 3125.000 3132.000 5390.000 5391.000 5459.000 5463.000 5472.000 5500.000 5502.000 5503.000 5507.000 5512.000 5526.000 5533.000 5578.000 5593.000 5614.000 5976.000 5980.000 6243.000 6245.000 6432.000 6541.000 6549.000 6590.000 6608.000 6640.000 6645.000 6663.000 6672.000 7784.000 7785.000 7799.000 7800.000 7832.000 7838.000 7843.000 7849.000
ENBL.AEXP	STRING	REFERENCED AT : 884.000 951.000 954.000 1481.000 1687.000 1688.000 1742.000 1836.000 1841.000 3488.000 3498.000 5115.000 5315.000 5316.000 7749.000 7761.000 7960.000 8025.000
ENBL55	STRING	REFERENCED AT : 1786.000 4424.000
ENTER.WCS	LABEL	REFERENCED AT : 5558.000 5575.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

```
ES                          LABEL    REFERENCED AT :   4480.000 8379.000

EVALUATE.PSD2               LABEL    REFERENCED AT :   1842.000 5321.000 5344.000

EVALUATE.PSD2.MAP           LABEL    REFERENCED AT :    969.000 2496.000 5350.000

EVENT.POLL                  LABEL    REFERENCED AT :    670.000 1630.000 7878.000 7938.000

EXCLUDE.CD.CLASSF           LABEL    REFERENCED AT :   2263.000 2434.000 2718.000

EXECUTE.ACK                 LABEL    REFERENCED AT :    821.000  823.000  828.000

EXECUTE.INTR.CTL            LABEL    REFERENCED AT :   2191.000 2200.000 2546.000

EXECUTE.LOAD.MAP            LABEL    REFERENCED AT :   6506.000 6582.000

EXFF                        STRING   REFERENCED AT :    639.000  683.000 1092.000 1097.000 5330.000 7954.000

EXFLAG                      STRING   REFERENCED AT :   1136.000 1758.000 4410.000 4420.000 5098.000 5141.000 5170.000 5173.000
                                     5186.000 5196.000 5258.000 5268.000 5271.000 7203.000

EXIT                        LABEL    REFERENCED AT :   5824.000 5853.000 5861.000 6300.000

EXIT.CLEAR                  LABEL    REFERENCED AT :   6545.000 6547.000

EXIT.LD.PSEUDO              LABEL    REFERENCED AT :   6644.000

EXIT.LOAD                   LABEL    REFERENCED AT :   6583.000 6595.000 6607.000

EXIT.USE.CURRENT            LABEL    REFERENCED AT :   6417.000 6428.000

EXM                         LABEL    REFERENCED AT :   5185.000 8136.000

EXM2                        LABEL    REFERENCED AT :   5186.000 5188.000 5193.000

EXM3                        LABEL    REFERENCED AT :   5189.000 5196.000

EXMR                        LABEL    REFERENCED AT :   5197.000 5200.000

EXPK                        REGISTER REFERENCED AT :   6753.000 6767.000 6771.000 6778.000 6779.000 6782.000 6788.000 6795.000
                                     6796.000 6814.000 6828.000 6832.000 6839.000 6840.000 6843.000 6849.000 6856.000 6857.000
                                     6877.000 6893.000 6897.000 6906.000 6907.000 6911.000 6916.000 6925.000 6926.000 6934.000
                                     6950.000 6954.000 6963.000 6964.000 6968.000 6974.000 6983.000 6984.000 6990.000 6991.000
                                     6996.000 7011.000 7018.000 7019.000 7046.000 7047.000 7052.000 7065.000 7078.000 7080.000
                                     7081.000 7091.000 7092.000 7099.000 7117.000 7126.000 7127.000 7135.000 7137.000 7144.000
                                     7157.000 7175.000 7179.000 7180.000

EXR                         LABEL    REFERENCED AT :   5170.000 8405.000

EXRR                        LABEL    REFERENCED AT :   5174.000 5177.000

EXT.BRI.RETURN              LABEL    REFERENCED AT :   7273.000 7289.000

EXT.CMD.AICT                LABEL    REFERENCED AT :   7386.000 7563.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

EXT.CMD.ARSTX	LABEL	REFERENCED AT : 7380.000 7491.000
EXT.CMD.ICT	LABEL	REFERENCED AT : 7401.000 7589.000
EXT.CMD.RSTX	LABEL	REFERENCED AT : 7395.000 7514.000
EXT.CMD.WDOT	LABEL	REFERENCED AT : 7407.000 7672.000
EXT.DACI.EXIT	LABEL	REFERENCED AT : 7614.000
EXT.DUD	LABEL	REFERENCED AT : 7497.000 7502.000 7509.000
EXT.DUD.1	LABEL	REFERENCED AT : 7520.000 7525.000
EXT.DUD.2	LABEL	REFERENCED AT : 7570.000 7574.000
EXT.DUD.3	LABEL	REFERENCED AT : 594.000 7583.000
EXT.DUD.4	LABEL	REFERENCED AT : 7534.000 7542.000
EXT.ICT.CK.TIMEOUT	LABEL	REFERENCED AT : 7603.000
EXT.ICT.PATCH	LABEL	REFERENCED AT : 7276.000 7596.000
EXT.IO.FLG	VARIABLE	REFERENCED AT : 527.000 7470.000 7666.000 7668.000
EXT.IO.INST	LABEL	REFERENCED AT : 7200.000 8581.000 8583.000
EXT.IO.REG.LOAD	LABEL	REFERENCED AT : 1903.000
EXT.IPL.ERROR	LABEL	REFERENCED AT : 7696.000 7731.000 7740.000 7756.000 7757.000
EXT.IPL.ERROR.1	LABEL	REFERENCED AT : 7728.000 7748.000 7749.000 7755.000
EXT.PHY.ADDR.0	LABEL	REFERENCED AT : 7426.000 7501.000
EXT.RAM.LOAD	LABEL	REFERENCED AT : 7422.000 7479.000 7551.000
EXT.RAM.LOADED	LABEL	REFERENCED AT : 7477.000 7480.000 7485.000 7552.000
EXT.RAM.LOADED1	LABEL	REFERENCED AT : 7549.000 7557.000
EXT.RSTX.RAM.LOAD	LABEL	REFERENCED AT : 7436.000 7529.000
EXT.TRANSFER.EXIT	LABEL	REFERENCED AT : 7391.000 7413.000
EXT.UPDATE.ACTIVE.COUNT	LABEL	REFERENCED AT : 7621.000
EXT.UPDATE.INTR.FLAGS	LABEL	REFERENCED AT : 7609.000 7619.000 7622.000 7625.000
EXTERNAL.IPU.TEST	LABEL	REFERENCED AT : 748.000 777.000 1018.000 7875.000 7937.000
EXTG	LABEL	REFERENCED AT : 650.000 4437.000 4502.000 4527.000 4542.000 4547.000 4575.000 5100.000 5143.000 5331.000 5442.000 7956.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

EXTG	STRING	REFERENCED AT : 651.000 671.000 5431.000
EXTG.NPM.ENTRY	LABEL	REFERENCED AT : 766.000 1006.000
EXTG.TFST	LABEL	REFERENCED AT : 702.000 738.000
EXTL	STRING	REFERENCED AT : 2531.000 3537.000 6729.000
EXTL.EXIT	LABEL	REFERENCED AT : 4108.000 4233.000 4244.000
EXTL.TFST	LABEL	REFERENCED AT : 700.000 764.000
EXTLW	STRING	REFERENCED AT : 2068.000 2430.000 3523.000 3975.000 4233.000 4244.000 4639.000 4851.000 4882.000 4971.000 5000.000 5132.000 5190.000 5216.000 5302.000 5456.000 5571.000 6339.000
EXTPROC	LABEL	REFERENCED AT : 659.000 2630.000 2636.000 5281.000
EXTPROC1	LABEL	REFERENCED AT : 2163.000 2163.000 2164.000 2518.000 2627.000 3525.000 3538.000 3975.000 4109.000 4640.000 4852.000 4883.000 4972.000 5001.000 5133.000 5190.000 5195.000 5217.000 5302.000 5456.000 5572.000 6747.000
FALSE	STRING	REFERENCED AT : 1091.000 1095.000 1678.000 1761.000 5097.000 5140.000 5175.000 5177.000 5198.000 5200.000 5253.000 5328.000 7856.000 7908.000
FAST.DEACTIVATE	LABEL	REFERENCED AT : 6388.000 6427.000
FAST.FPEXIT	LABEL	REFERENCED AT : 3645.000 3647.000 3663.000 3717.000 3718.000 4143.000
FAST.NORM	LABEL	REFERENCED AT : 3633.000 3642.000
FC1V	STRING	REFERENCED AT : 3582.000 3658.000 3841.000 5095.000 5137.000 5174.000 5197.000 6732.000 6816.000 6880.000 6937.000
FETCH	LABEL	REFERENCED AT : 5163.000
FETCH.DEFAULT.IPL	LABEL	REFERENCED AT : 1282.000
FETCH.MSCD	LABEL	REFERENCED AT : 6465.000 6480.000 6563.000
FETCH.MSD1	LABEL	REFERENCED AT : 6470.000 6485.000
FETCH.MSW	LABEL	REFERENCED AT : 4655.000 4685.000 4812.000
FETCH.RETURN	LABEL	REFERENCED AT : 1500.000 2168.000 2169.000 3489.000 3499.000 3505.000 3568.000 3786.000 3828.000 4401.000 4476.000 4481.000 4482.000 4487.000 4488.000 4523.000 4531.000 4579.000 4591.000 4603.000 4619.000 4768.000 5053.000 5392.000 5403.000 5412.000 5537.000 6684.000 7993.000 8039.000
FETCHPC	STRING	REFERENCED AT : 1087.000 1090.000 1094.000 1823.000 2279.000 2413.000 2500.000 2716.000 3479.000 3648.000 4493.000 4499.000 4508.000 4514.000 4515.000 4517.000 4519.000 4520.000 4524.000 4537.000 4539.000 4544.000 4552.000 4554.000 4572.000 4656.000 4659.000 4671.000 4686.000 4707.000 4710.000 4726.000 4729.000 4746.000 4752.000 4795.000 4899.000 4928.000 4979.000 5003.000 5028.000 5039.000 5068.000 5095.000 5146.000 5163.000 5197.000 5253.000 5279.000 5323.000 5325.000 5472.000 5577.000 6763.000 6767.000 6785.000 6824.000 6828.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                    6846.000 6897.000 6901.000 6907.000 6916.000 6920.000 6926.000 6954.000 6958.000 6964.000
                                    6974.000 6978.000 6984.000 7008.000 7070.000 7083.000 7117.000 7121.000 7127.000 7175.000
                                    7180.000 7184.000 7354.000 7952.000

FETCHV                      STRING  REFERENCED AT :   2524.000 2532.000 5080.000 5090.000 5111.000 5125.000 5134.000 5137.000
                                    5188.000 5192.000 5206.000

FFINT                       STRING  REFERENCED AT :    788.000 2023.000 2132.000 2211.000 2429.000 2516.000 3165.000 4435.000
                                    7702.000 7716.000

FFPADDD                     LABEL   REFERENCED AT :   6874.000 8475.000 8650.000

FFPADDS                     LABEL   REFERENCED AT :   6751.000 8452.000 8646.000

FFPCC                       STRING  REFERENCED AT :   8444.000 8445.000 8452.000 8453.000 8467.000 8468.000 8475.000 8476.000
                                    8638.000 8642.000 8646.000 8650.000 8661.000 8665.000 8669.000 8673.000

FFPDVDD                     LABEL   REFERENCED AT :   7132.000 8468.000 8665.000

FFPDVDS                     LABEL   REFERENCED AT :   7044.000 8445.000 8661.000

FFPMPYD                     LABEL   REFERENCED AT :   7088.000 8476.000 8673.000

FFPMPYS                     LABEL   REFERENCED AT :   6988.000 8453.000 8669.000

FFPSUBD                     LABEL   REFERENCED AT :   6931.000 8467.000 8642.000

FFPSUBS                     LABEL   REFERENCED AT :   6812.000 8444.000 8638.000

FFRUN                       STRING  REFERENCED AT :    676.000  677.000 5437.000 5729.000 5973.000 5992.000 6006.000 6088.000
                                    6147.000 6214.000 6270.000 6307.000

FFSYSR                      STRING  REFERENCED AT :   1011.000 1105.000

FILL.PIPELINE               LABEL   REFERENCED AT :    789.000  922.000 1082.000 1330.000

FILLEXP                     STRING  REFERENCED AT :   3582.000 3585.000 3587.000 3610.000 3716.000 3846.000 3851.000 3889.000
                                    3896.000 3919.000 4009.000 4066.000 4075.000

FINAL.FLG                   VARIABLE REFERENCED AT :   529.000  594.000 3025.000 3033.000 7279.000 7441.000 7519.000 7525.000
                                    7534.000 7605.000 7668.000

FINAL.ON.BUS                LABEL   REFERENCED AT :   7396.000 7402.000 7409.000

FIRMWARE.FP                 LABEL   REFERENCED AT :   3515.000 7170.000 8640.000 8644.000 8648.000 8652.000 8663.000 8667.000
                                    8671.000 8675.000

FIRST.IPL.IO                LABEL   REFERENCED AT :   1295.000 1313.000 1315.000

FIX.FP                      LABEL   REFERENCED AT :   3644.000 3650.000

FIXDV.AE                    LABEL   REFERENCED AT :   3771.000 3779.000 3787.000

FIXED.DIV                   LABEL   REFERENCED AT :   3751.000 3752.000 3758.000 3760.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

FIXED.MPY	LABEL	REFERENCED AT : 3737.000 3738.000 3745.000 3792.000
FIXED.MPY.NEG.RESULT	LABEL	REFERENCED AT : 3504.000 3573.000
FIXED.MPY.POS.RESULT	LABEL	REFERENCED AT : 3571.000 3827.000
FIXEXP	STRING	REFERENCED AT : 3648.000
FLAG	STRING	REFERENCED AT : 410.000 1293.000 1322.000 2969.000 3269.000 3274.000 3385.000 3731.000 3736.000 3743.000 3833.000 3834.000 4470.000 4473.000 4475.000 5296.000 5329.000 5535.000 5536.000 5540.000 5545.000 7685.000 7686.000 7900.000
FLAG.EQUAL.MAPMODE	LABEL	REFERENCED AT : 3383.000
FORCE.PROG.VIOL	LABEL	REFERENCED AT : 2339.000
FORCEZ	STRING	REFERENCED AT : 2426.000 3914.000 4026.000 4192.000 4940.000 6586.000 6879.000 6936.000 7093.000 7138.000 7653.000
FORMAT.75.AEXP	LABEL	REFERENCED AT : 1480.000 3306.000
FORMAT.75.AEXP.75MODE	LABEL	REFERENCED AT : 1476.000 1613.000
FORMAT.ACK.ACTIVATE	LABEL	REFERENCED AT : 7454.000 7457.000
FORMAT.ACK.DEACTIVATE	LABEL	REFERENCED AT : 7455.000
FORMAT.TCW	LABEL	REFERENCED AT : 2038.000 2351.000 2351.000 2464.000 2464.000
FP.CURRENT.EXIT	LABEL	REFERENCED AT : 6729.000 6746.000
FP.DOUBLE.CHECK	LABEL	REFERENCED AT : 6744.000 6881.000 6938.000 7096.000 7141.000
FP.EXTPROC1	LABEL	REFERENCED AT : 3524.000
FP.INDIRECT	LABEL	REFERENCED AT : 6726.000 6730.000 6754.000 6815.000 6878.000 6935.000 6992.000 7048.000 7090.000 7136.000
FP1.0	LABEL	REFERENCED AT : 3517.000 3523.000
FPDW.CLROP1	LABEL	REFERENCED AT : 4039.000 4054.000
FPDW.CLROP2	LABEL	REFERENCED AT : 4063.000 4072.000
FPDW.DELAYED.EXIT	LABEL	REFERENCED AT : 4122.000 4124.000 4131.000 4135.000
FPDW.DIV0	LABEL	REFERENCED AT : 3933.000
FPDW.EXIT	LABEL	REFERENCED AT : 4020.000 4137.000
FPDW.EXP1GT	LABEL	REFERENCED AT : 4031.000 4059.000
FPDW.MPY.TAB	LABEL	REFERENCED AT : 4193.000 4235.000 4283.000
FPDW.NORM.1	LABEL	REFERENCED AT : 4101.000 4120.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

FPDW.NORM.2	LABEL	REFERENCED AT : 4017.000 4104.000
FPDW.NORM.3	LABEL	REFERENCED AT : 4019.000 4103.000 4118.000 4129.000
FPDW.NORMALIZE	LABEL	REFERENCED AT : 4001.000 4004.000 4080.000 4085.000 4086.000 4267.000 4271.000
FPDW.OFUF.TEST	LABEL	REFERENCED AT : 3526.000 4139.000
FPDW.OPR1	LABEL	REFERENCED AT : 4043.000 4055.000
FPDW.OPR1EQOPR2	LABEL	REFERENCED AT : 4037.000 4048.000
FPDW.REENTER.NORM	LABEL	REFERENCED AT : 4079.000 4153.000
FPDW.SCALED	LABEL	REFERENCED AT : 4050.000 4068.000 4073.000 4074.000
FPDW.SET.RESULT.ZERO	LABEL	REFERENCED AT : 4146.000 4186.000 4199.000
FPDW.SHIFT	LABEL	REFERENCED AT : 4095.000 4144.000
FPDW.ZERO	LABEL	REFERENCED AT : 4123.000 4147.000
FPDWORD	STRING	REFERENCED AT : 6719.000 6875.000 6932.000 7037.000 7089.000 7133.000 7169.000
FPSP.FIX.DI	LABEL	REFERENCED AT : 3599.000 3602.000
FPSP.FIX.N	LABEL	REFERENCED AT : 3586.000 3594.000
FPSW.MPY	LABEL	REFERENCED AT : 3901.000 4364.000
FR(	STRING	REFERENCED AT : 603.000 652.000 652.000 674.000 694.000 695.000 699.000 719.000 719.000 744.000 744.000 767.000 767.000 770.000 770.000 775.000 775.000 814.000 820.000 858.000 865.000 905.000 908.000 909.000 919.000 919.000 927.000 927.000 932.000 934.000 955.000 998.000 998.000 1005.000 1005.000 1027.000 1028.000 1045.000 1050.000 1050.000 1062.000 1062.000 1086.000 1103.000 1104.000 1111.000 1113.000 1124.000 1125.000 1137.000 1137.000 1168.000 1169.000 1192.000 1195.000 1203.000 1205.000 1208.000 1215.000 1219.000 1224.000 1255.000 1258.000 1258.000 1269.000 1290.000 1290.000 1291.000 1292.000 1299.000 1352.000 1464.000 1469.000 1471.000 1471.000 1515.000 1643.000 1676.000 1682.000 1682.000 1714.000 1748.000 1748.000 1752.000 1755.000 1755.000 1762.000 1762.000 1807.000 1810.000 1894.000 1896.000 1901.000 1928.000 2039.000 2076.000 2086.000 2088.000 2088.000 2100.000 2180.000 2228.000 2237.000 2248.000 2262.000 2263.000 2274.000 2290.000 2400.000 2400.000 2460.000 2520.000 2537.000 2538.000 2538.000 2551.000 2552.000 2576.000 2576.000 2587.000 2587.000 2597.000 2597.000 2608.000 2609.000 2609.000 2620.000 2621.000 2621.000 2629.000 2629.000 2634.000 2634.000 2639.000 2639.000 2660.000 2664.000 2667.000 2688.000 2699.000 2727.000 2732.000 2741.000 2742.000 2745.000 2746.000 3097.000 3097.000 3124.000 3124.000 3146.000 3146.000 3155.000 3166.000 3172.000 3173.000 3173.000 3209.000 3209.000 3214.000 3214.000 3230.000 3273.000 3273.000 3305.000 3316.000 3382.000 3406.000 3422.000 3423.000 4776.000 4908.000 5112.000 5233.000 5240.000 5242.000 5247.000 5247.000 5259.000 5259.000 5275.000 5277.000 5280.000 5305.000 5416.000 5416.000 5429.000 5429.000 5439.000 5439.000 5501.000 5506.000 6091.000 6098.000 6342.000 6342.000 6389.000 6466.000 6467.000 6468.000 6501.000 6543.000 6589.000 6589.000 6595.000 6625.000 6670.000 6680.000 7232.000 7243.000 7255.000 7255.000 7271.000 7271.000 7273.000 7295.000 7298.000 7301.000 7307.000 7310.000 7316.000 7316.000 7321.000 7324.000 7324.000 7327.000 7327.000 7330.000 7330.000 7333.000 7333.000 7335.000 7335.000 7348.000 7423.000 7425.000 7425.000 7439.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

7439.000 7461.000 7481.000 7553.000 7617.000 7637.000 7651.000 7651.000 7656.000 7656.000
7663.000 7674.000 7690.000 7693.000 7693.000 7750.000 7883.000 7952.000 8041.000 8041.000

FRCWORD STRING REFERENCED AT : 885.000 943.000 951.000 963.000 1110.000 1510.000 1838.000 1900.000
1906.000 1933.000 2015.000 2369.000 2397.000 2477.000 2524.000 2532.000 2665.000 2690.000
3212.000 3226.000 3309.000 3315.000 3319.000 3322.000 3326.000 3411.000 3418.000 3516.000
3520.000 3534.000 4472.000 4782.000 4813.000 4822.000 4830.000 4895.000 4912.000 4945.000
4963.000 4967.000 4975.000 4981.000 4992.000 4996.000 5080.000 5090.000 5111.000 5125.000
5134.000 5137.000 5188.000 5192.000 5206.000 5296.000 5300.000 5307.000 5454.000 5503.000
5509.000 5523.000 5526.000 5892.000 5895.000 6474.000 6482.000 6516.000 6566.000 6693.000
6729.000 6752.000 6813.000 6875.000 6932.000 6989.000 7045.000 7089.000 7134.000 7530.000
7600.000 7839.000

FULLMAR STRING REFERENCED AT : 410.000 714.000 739.000 768.000 814.000 824.000 870.000 929.000
934.000 942.000 950.000 960.000 1084.000 1113.000 1151.000 1181.000 1185.000 1236.000
1270.000 1295.000 1334.000 1335.000 1337.000 1342.000 1353.000 1356.000 1429.000 1616.000
1620.000 1678.000 1713.000 1723.000 1810.000 1812.000 1835.000 1899.000 1905.000 1932.000
2028.000 2068.000 2125.000 2163.000 2167.000 2202.000 2289.000 2351.000 2364.000 2381.000
2396.000 2423.000 2426.000 2464.000 2476.000 2493.000 2499.000 2550.000 2637.000 2663.000
2689.000 2764.000 2766.000 2771.000 2774.000 2777.000 2837.000 2841.000 2937.000 2956.000
3100.000 3119.000 3211.000 3225.000 3308.000 3314.000 3318.000 3321.000 3325.000 3348.000
3349.000 3417.000 3530.000 3644.000 3646.000 3714.000 3720.000 3987.000 4138.000 4218.000
4220.000 4249.000 4265.000 4471.000 4766.000 5353.000 5827.000 5911.000 5932.000 5935.000
5946.000 5975.000 6026.000 6031.000 6115.000 6149.000 6209.000 6301.000 6471.000 6481.000
6515.000 6545.000 6549.000 6565.000 6585.000 6590.000 6599.000 6627.000 6642.000 6658.000
6682.000 7491.000 7528.000 7563.000 7597.000 7650.000 7653.000 7686.000 7691.000 7699.000
7713.000 7742.000 7838.000 7902.000

GO TO STRING REFERENCED AT : 672.000 673.000 677.000 678.000 693.000 696.000 697.000 698.000
700.000 701.000 702.000 705.000 712.000 714.000 716.000 718.000 722.000 723.000
739.000 741.000 743.000 746.000 748.000 751.000 765.000 772.000 777.000 789.000
801.000 814.000 815.000 818.000 824.000 828.000 833.000 835.000 840.000 842.000
849.000 854.000 866.000 874.000 907.000 910.000 922.000 931.000 933.000 938.000
946.000 957.000 964.000 970.000 971.000 1004.000 1006.000 1009.000 1010.000 1011.000
1012.000 1015.000 1037.000 1038.000 1041.000 1044.000 1047.000 1048.000 1052.000 1060.000
1064.000 1076.000 1089.000 1105.000 1113.000 1114.000 1116.000 1120.000 1121.000 1123.000
1129.000 1134.000 1135.000 1136.000 1156.000 1186.000 1238.000 1255.000 1259.000 1265.000
1273.000 1279.000 1281.000 1303.000 1306.000 1307.000 1311.000 1312.000 1315.000 1320.000
1323.000 1325.000 1330.000 1346.000 1349.000 1359.000 1364.000 1367.000 1368.000 1370.000
1372.000 1375.000 1392.000 1395.000 1399.000 1408.000 1440.000 1441.000 1444.000 1445.000
1448.000 1449.000 1451.000 1461.000 1462.000 1475.000 1481.000 1500.000 1514.000 1518.000
1519.000 1523.000 1524.000 1537.000 1552.000 1553.000 1554.000 1557.000 1576.000 1580.000
1582.000 1583.000 1585.000 1594.000 1608.000 1611.000 1616.000 1622.000 1627.000 1630.000
1645.000 1646.000 1651.000 1652.000 1653.000 1657.000 1665.000 1681.000 1685.000 1691.000
1692.000 1703.000 1716.000 1727.000 1732.000 1735.000 1742.000 1749.000 1756.000 1757.000
1758.000 1760.000 1769.000 1770.000 1803.000 1844.000 1897.000 1902.000 1911.000 1940.000
1950.000 1955.000 1975.000 2009.000 2023.000 2025.000 2028.000 2029.000 2045.000 2046.000
2052.000 2053.000 2066.000 2068.000 2070.000 2079.000 2080.000 2089.000 2090.000 2092.000
2099.000 2101.000 2121.000 2127.000 2128.000 2130.000 2132.000 2133.000 2134.000 2137.000
2139.000 2142.000 2144.000 2148.000 2150.000 2152.000 2153.000 2163.000 2164.000 2168.000
2169.000 2180.000 2185.000 2186.000 2189.000 2193.000 2194.000 2202.000 2206.000 2207.000
2209.000 2211.000 2212.000 2213.000 2217.000 2220.000 2225.000 2226.000 2230.000 2231.000
2233.000 2235.000 2239.000 2240.000 2245.000 2257.000 2270.000 2271.000 2272.000 2277.000
2293.000 2294.000 2296.000 2300.000 2301.000 2317.000 2326.000 2331.000 2338.000 2341.000
2344.000 2364.000 2372.000 2382.000 2408.000 2410.000 2411.000 2415.000 2428.000 2429.000

SEL 32/75 CPU

MICROCODE CROSS-REFERENCE LIST

2433.000 2436.000 2437.000 2448.000 2449.000 2451.000 2453.000 2454.000 2457.000 2459.000
2462.000 2472.000 2478.000 2492.000 2516.000 2518.000 2519.000 2522.000 2523.000 2526.000
2531.000 2533.000 2539.000 2544.000 2545.000 2547.000 2552.000 2564.000 2567.000 2578.000
2599.000 2610.000 2612.000 2622.000 2625.000 2630.000 2636.000 2637.000 2668.000 2672.000
2673.000 2679.000 2682.000 2688.000 2692.000 2693.000 2718.000 2719.000 2720.000 2723.000
2729.000 2734.000 2735.000 2743.000 2744.000 2747.000 2748.000 2755.000 2756.000 2765.000
2770.000 2773.000 2776.000 2778.000 2781.000 2789.000 2797.000 2807.000 2809.000 2815.000
2816.000 2827.000 2828.000 2836.000 2840.000 2842.000 2850.000 2851.000 2862.000 2863.000
2872.000 2873.000 2880.000 2881.000 2890.000 2900.000 2918.000 2928.000 2958.000 2962.000
2971.000 3010.000 3082.000 3103.000 3122.000 3130.000 3165.000 3174.000 3181.000 3218.000
3222.000 3229.000 3232.000 3267.000 3268.000 3316.000 3327.000 3351.000 3355.000 3358.000
3359.000 3385.000 3409.000 3421.000 3425.000 3427.000 3442.000 3450.000 3451.000 3452.000
3460.000 3461.000 3463.000 3472.000 3475.000 3476.000 3487.000 3489.000 3497.000 3499.000
3505.000 3522.000 3523.000 3525.000 3532.000 3535.000 3537.000 3538.000 3544.000 3547.000
3548.000 3552.000 3554.000 3555.000 3557.000 3558.000 3561.000 3568.000 3582.000 3585.000
3594.000 3595.000 3599.000 3600.000 3601.000 3622.000 3631.000 3633.000 3635.000 3637.000
3638.000 3641.000 3644.000 3645.000 3646.000 3651.000 3658.000 3661.000 3663.000 3680.000
3683.000 3687.000 3691.000 3696.000 3699.000 3702.000 3705.000 3706.000 3708.000 3711.000
3716.000 3717.000 3718.000 3720.000 3723.000 3729.000 3730.000 3737.000 3738.000 3744.000
3745.000 3751.000 3752.000 3757.000 3758.000 3771.000 3774.000 3779.000 3781.000 3783.000
3786.000 3790.000 3797.000 3809.000 3818.000 3825.000 3828.000 3837.000 3841.000 3879.000
3881.000 3891.000 3898.000 3899.000 3904.000 3907.000 3926.000 3943.000 3963.000 3967.000
3969.000 3970.000 3975.000 3977.000 3981.000 3985.000 4001.000 4004.000 4030.000 4031.000
4037.000 4038.000 4039.000 4050.000 4055.000 4062.000 4063.000 4068.000 4073.000 4080.000
4085.000 4095.000 4097.000 4107.000 4109.000 4111.000 4124.000 4138.000 4139.000 4142.000
4143.000 4145.000 4147.000 4153.000 4186.000 4199.000 4233.000 4244.000 4248.000 4264.000
4267.000 4268.000 4271.000 4273.000 4286.000 4287.000 4288.000 4289.000 4300.000 4301.000
4304.000 4365.000 4390.000 4401.000 4409.000 4413.000 4453.000 4458.000 4469.000 4531.000
4570.000 4630.000 4637.000 4639.000 4640.000 4649.000 4670.000 4682.000 4695.000 4696.000
4704.000 4723.000 4741.000 4750.000 4789.000 4818.000 4827.000 4835.000 4844.000 4849.000
4850.000 4851.000 4852.000 4861.000 4869.000 4880.000 4882.000 4883.000 4892.000 4897.000
4922.000 4936.000 4944.000 4946.000 4948.000 4953.000 4962.000 4969.000 4971.000 4972.000
4982.000 4991.000 4993.000 4998.000 4999.000 5000.000 5001.000 5053.000 5078.000 5091.000
5112.000 5124.000 5187.000 5190.000 5194.000 5195.000 5211.000 5212.000 5216.000 5217.000
5225.000 5247.000 5251.000 5255.000 5257.000 5261.000 5269.000 5274.000 5297.000 5302.000
5303.000 5311.000 5326.000 5331.000 5345.000 5356.000 5370.000 5373.000 5389.000 5392.000
5394.000 5403.000 5412.000 5414.000 5418.000 5431.000 5432.000 5433.000 5434.000 5435.000
5437.000 5442.000 5450.000 5452.000 5456.000 5458.000 5460.000 5542.000 5551.000 5555.000
5568.000 5580.000 5618.000 5728.000 5729.000 5731.000 5735.000 5736.000 5738.000 5739.000
5750.000 5760.000 5768.000 5769.000 5770.000 5771.000 5772.000 5773.000 5774.000 5799.000
5802.000 5804.000 5805.000 5812.000 5814.000 5815.000 5816.000 5817.000 5822.000 5824.000
5835.000 5840.000 5843.000 5853.000 5861.000 5873.000 5879.000 5886.000 5888.000 5890.000
5893.000 5901.000 5926.000 5928.000 5934.000 5935.000 5946.000 5947.000 5949.000 5956.000
5965.000 5966.000 5967.000 5968.000 5969.000 5970.000 5971.000 5973.000 5984.000 5987.000
5990.000 5992.000 5996.000 6003.000 6006.000 6010.000 6017.000 6021.000 6028.000 6029.000
6035.000 6039.000 6041.000 6042.000 6045.000 6053.000 6057.000 6068.000 6077.000 6083.000
6088.000 6092.000 6097.000 6102.000 6108.000 6111.000 6118.000 6121.000 6123.000 6128.000
6130.000 6137.000 6144.000 6147.000 6153.000 6155.000 6162.000 6174.000 6189.000 6195.000
6202.000 6208.000 6210.000 6212.000 6214.000 6216.000 6230.000 6238.000 6246.000 6254.000
6260.000 6263.000 6269.000 6270.000 6287.000 6292.000 6299.000 6307.000 6308.000 6311.000
6317.000 6328.000 6330.000 6337.000 6339.000 6341.000 6361.000 6383.000 6394.000 6397.000
6401.000 6403.000 6405.000 6408.000 6410.000 6413.000 6420.000 6422.000 6427.000 6433.000
6471.000 6476.000 6485.000 6498.000 6516.000 6545.000 6546.000 6553.000 6568.000 6583.000
6588.000 6643.000 6659.000 6664.000 6684.000 6692.000 6694.000 6695.000 6696.000 6703.000
6720.000 6730.000 6747.000 6754.000 6755.000 6763.000 6815.000 6816.000 6824.000 6878.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

		6880.000 6889.000 6935.000 6937.000 6946.000 6992.000 6994.000 7001.000 7048.000 7050.000
		7059.000 7072.000 7090.000 7095.000 7104.000 7136.000 7140.000 7151.000 7165.000 7201.000
		7202.000 7204.000 7205.000 7238.000 7273.000 7275.000 7278.000 7288.000 7290.000 7292.000
		7295.000 7298.000 7301.000 7304.000 7307.000 7310.000 7314.000 7317.000 7319.000 7322.000
		7345.000 7348.000 7357.000 7358.000 7391.000 7426.000 7435.000 7437.000 7454.000 7461.000
		7464.000 7471.000 7477.000 7492.000 7494.000 7495.000 7500.000 7502.000 7505.000 7507.000
		7508.000 7509.000 7515.000 7517.000 7518.000 7524.000 7529.000 7533.000 7541.000 7549.000
		7564.000 7565.000 7567.000 7568.000 7573.000 7574.000 7577.000 7578.000 7580.000 7581.000
		7590.000 7592.000 7593.000 7596.000 7599.000 7609.000 7613.000 7618.000 7619.000 7639.000
		7655.000 7659.000 7661.000 7662.000 7664.000 7670.000 7673.000 7696.000 7702.000 7703.000
		7704.000 7728.000 7731.000 7740.000 7748.000 7749.000 7753.000 7759.000 7762.000 7767.000
		7809.000 7819.000 7822.000 7829.000 7834.000 7836.000 7840.000 7842.000 7844.000 7850.000
		7950.000 7956.000 7985.000 7989.000 7991.000 7993.000 8010.000 8016.000 8019.000 8020.000
		8022.000 8034.000 8039.000 8052.000 8053.000 8054.000 8055.000 8069.000
GOTO	STRING	REFERENCED AT : 603.000 674.000 817.000 979.000 980.000 986.000 1018.000 1109.000
		1492.000 1782.000 1794.000 1795.000 1797.000 1805.000 1809.000 1850.000 1851.000 1857.000
		1858.000 1867.000 2181.000 2267.000 2269.000 2430.000 2512.000 2527.000 3128.000 3143.000
		3157.000 3300.000 3407.000 4019.000 4020.000 4103.000 4104.000 4118.000 4120.000 4131.000
		4397.000 4431.000 4437.000 4438.000 4443.000 4452.000 4465.000 4476.000 4481.000 4482.000
		4487.000 4488.000 4502.000 4512.000 4535.000 4542.000 4547.000 4559.000 4562.000 4567.000
		4575.000 4579.000 4583.000 4588.000 4590.000 4591.000 4594.000 4596.000 4599.000 4600.000
		4601.000 4603.000 4604.000 4606.000 4610.000 4613.000 4619.000 4760.000 4763.000 4768.000
		5014.000 5024.000 5069.000 5077.000 5079.000 5081.000 5089.000 5092.000 5100.000 5116.000
		5126.000 5130.000 5132.000 5133.000 5135.000 5143.000 5154.000 5159.000 5160.000 5162.000
		5171.000 5174.000 5185.000 5189.000 5233.000 5234.000 5238.000 5252.000 5273.000 5281.000
		5425.000 5490.000 5504.000 5505.000 5510.000 5513.000 5531.000 5537.000 5544.000 5566.000
		5571.000 5572.000 5575.000 6441.000 6443.000 6450.000 6452.000 6595.000 6740.000 6742.000
		6805.000 6806.000 6808.000 6867.000 6868.000 6869.000 6870.000 7170.000 7344.000 7861.000
		7866.000 7867.000 7868.000 7872.000 7877.000 7878.000 7886.000 7891.000 7892.000 7897.000
		7898.000 7899.000 7903.000 7914.000 7927.000 7929.000 7931.000 7937.000 7938.000
GRIO	LABEL	REFERENCED AT : 7308.000
HALT	LABEL	REFERENCED AT : 5388.000 8375.000
HALT	STRING	REFERENCED AT : 1030.000 5391.000
HANDLE.75.INT	LABEL	REFERENCED AT : 856.000 6381.000
HANDLE.75.PSW	LABEL	REFERENCED AT : 897.000 940.000
HIO	LABEL	REFERENCED AT : 7305.000
HIREG	STRING	REFERENCED AT : 787.000 865.000 1035.000 1083.000 1100.000 1132.000 1150.000 1270.000
		1288.000 1464.000 1471.000 1534.000 1570.000 1577.000 1629.000 1655.000 1708.000 1713.000
		1745.000 1781.000 1823.000 2167.000 2279.000 2413.000 2524.000 2551.000 2716.000 3065.000
		3171.000 3179.000 3205.000 3212.000 3314.000 3348.000 3418.000 3516.000 3649.000 3769.000
		3780.000 3796.000 4442.000 4459.000 4524.000 5239.000 5577.000 5734.000 6002.000 6301.000
		7355.000 7750.000 7812.000 7952.000
HOP	STRING	REFERENCED AT : 51.000 94.000 210.000 594.000 671.000 675.000 749.000 769.000
		774.000 788.000 821.000 826.000 839.000 846.000 847.000 871.000 872.000 924.000
		944.000 948.000 968.000 997.000 1013.000 1016.000 1017.000 1023.000 1036.000 1272.000
		1276.000 1316.000 1322.000 1477.000 1512.000 1573.000 1690.000 1702.000 1734.000 1768.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                1777.000 1785.000 1786.000 1817.000 1818.000 1819.000 1891.000 1892.000 1951.000 1964.000
                                1984.000 1991.000 1998.000 2005.000 2015.000 2049.000 2050.000 2078.000 2126.000 2140.000
                                2141.000 2151.000 2205.000 2216.000 2218.000 2224.000 2409.000 2458.000 2491.000 2587.000
                                2597.000 2635.000 2979.000 2980.000 2999.000 3000.000 3002.000 3003.000 3005.000 3006.000
                                3026.000 3027.000 3030.000 3034.000 3053.000 3054.000 3056.000 3057.000 3059.000 3060.000
                                3062.000 3095.000 3105.000 3108.000 3110.000 3115.000 3127.000 3135.000 3147.000 3227.000
                                3419.000 3521.000 3570.000 3685.000 3833.000 4122.000 4133.000 4219.000 4412.000 4422.000
                                4424.000 4435.000 4436.000 4457.000 4473.000 4493.000 4508.000 4514.000 4515.000 4517.000
                                4519.000 4520.000 4537.000 4558.000 4566.000 4568.000 4577.000 4585.000 4586.000 4598.000
                                4636.000 4659.000 4710.000 4729.000 4752.000 4762.000 4764.000 4779.000 4785.000 4790.000
                                4848.000 4879.000 4896.000 4907.000 4909.000 4918.000 4923.000 4943.000 4950.000 5023.000
                                5048.000 5068.000 5090.000 5095.000 5127.000 5128.000 5136.000 5137.000 5170.000 5186.000
                                5193.000 5197.000 5243.000 5258.000 5275.000 5277.000 5300.000 5301.000 5309.000 5354.000
                                5355.000 5358.000 5402.000 5411.000 5466.000 5535.000 5541.000 5543.000 5749.000 5777.000
                                5779.000 5781.000 5783.000 5785.000 5851.000 5898.000 5903.000 5904.000 5905.000 5906.000
                                5912.000 5997.000 6232.000 6276.000 6278.000 6280.000 6282.000 6340.000 6342.000 6344.000
                                6346.000 6348.000 6364.000 6366.000 6372.000 6396.000 6399.000 6407.000 6412.000 6417.000
                                6418.000 6497.000 6535.000 6587.000 6658.000 6729.000 6993.000 7049.000 7065.000 7139.000
                                7157.000 7203.000 7234.000 7277.000 7325.000 7328.000 7331.000 7381.000 7396.000 7402.000
                                7430.000 7440.000 7456.000 7486.000 7525.000 7531.000 7534.000 7558.000 7604.000 7622.000
                                7666.000 7716.000 7756.000 7818.000 7827.000 7846.000 7865.000 7896.000 8023.000 8059.000

HWDCNT                  STRING  REFERENCED AT :  6782.000 6843.000

HWORD                   STRING  REFERENCED AT :   705.000 2046.000 3737.000 3751.000 4659.000 4729.000 4750.000 4948.000

HWS                     STRING  REFERENCED AT :   603.000  672.000  678.000  816.000  859.000  897.000  970.000 1004.000
                                1064.000 1382.000 1389.000 1390.000 1405.000 1406.000 1520.000 1593.000 1664.000 1811.000
                                1814.000 1928.000 2027.000 2042.000 2310.000 2471.000 2677.000 2710.000 2886.000 3411.000
                                3994.000 3998.000 5158.000 5871.000 6171.000 6185.000 6480.000 6654.000 6672.000 6677.000
                                7250.000 7414.000

I0                      REGISTER REFERENCED AT :  1096.000 1097.000 1756.000 2266.000 2268.000 2270.000 2271.000 2293.000
                                2294.000 2300.000 2301.000 2308.000 2324.000 2343.000 2362.000 2522.000 2718.000 2720.000
                                2734.000 2735.000 2743.000 2744.000 2747.000 2748.000 2755.000 2815.000 2850.000 2880.000
                                3406.000 3424.000 3450.000 3460.000 3472.000 3474.000 3530.000 3533.000 3542.000 3745.000
                                3758.000 4409.000 4442.000 4553.000 4555.000 4558.000 4566.000 4630.000 4777.000 4869.000
                                4906.000 4962.000 4991.000 5013.000 5022.000 5023.000 5036.000 5037.000 5061.000 5089.000
                                5124.000 5144.000 5162.000 5170.000 5174.000 5186.000 5268.000 5271.000 5279.000 5329.000
                                5462.000 6719.000 6868.000 7169.000 7205.000 7206.000 7233.000 7954.000 8069.000

I1                      STRING  REFERENCED AT :  1135.000 1235.000 5146.000 5173.000 5175.000 5177.000

I1TOI0                  STRING  REFERENCED AT :  1083.000 1091.000 1095.000 2526.000 5081.000 5092.000 5097.000 5112.000
                                5136.000 5140.000 5145.000 5172.000 5174.000 5196.000 5198.000 5200.000 5253.000 5324.000
                                5328.000 7954.000

ICL.IOCD1               VARIABLE REFERENCED AT :   290.000 1300.000 1332.000 1355.000

ICL.IOCD2               VARIABLE REFERENCED AT :   254.000 1301.000 1333.000 1358.000 7690.000 7701.000

ICT                     LABEL   REFERENCED AT :  7578.000 7586.000

ICT                     STRING  REFERENCED AT :  2005.000 7402.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

ICT.CHECK	LABEL	REFERENCED AT : 6394.000 6411.000
ICT.DEACT	LABEL	REFERENCED AT : 6407.000 6412.000 6414.000 6420.000
ICT.DONE	LABEL	REFERENCED AT : 2218.000 2221.000
ICT.DRT	LABEL	REFERENCED AT : 7278.000 7590.000 7595.000
ICT.RETRY	LABEL	REFERENCED AT : 7588.000 7593.000
ICT.TRANSFER.OK	LABEL	REFERENCED AT : 7604.000 7608.000
IF	STRING	REFERENCED AT : 51.000 94.000 210.000 603.000 639.000 651.000 671.000 673.000 675.000 676.000 677.000 683.000 693.000 696.000 697.000 698.000 700.000 701.000 705.000 712.000 714.000 716.000 718.000 722.000 739.000 741.000 743.000 746.000 748.000 749.000 765.000 769.000 772.000 774.000 777.000 788.000 801.000 814.000 817.000 821.000 824.000 826.000 828.000 835.000 839.000 840.000 846.000 847.000 849.000 856.000 860.000 866.000 871.000 872.000 884.000 897.000 907.000 910.000 921.000 926.000 931.000 933.000 944.000 948.000 954.000 957.000 964.000 966.000 968.000 970.000 979.000 996.000 997.000 1004.000 1009.000 1011.000 1012.000 1013.000 1016.000 1017.000 1023.000 1036.000 1044.000 1047.000 1048.000 1063.000 1073.000 1075.000 1084.000 1085.000 1087.000 1089.000 1091.000 1095.000 1105.000 1109.000 1113.000 1120.000 1134.000 1135.000 1136.000 1156.000 1186.000 1255.000 1259.000 1272.000 1276.000 1279.000 1306.000 1311.000 1312.000 1315.000 1316.000 1322.000 1324.000 1346.000 1349.000 1364.000 1367.000 1368.000 1370.000 1377.000 1392.000 1395.000 1401.000 1439.000 1440.000 1441.000 1444.000 1445.000 1448.000 1449.000 1461.000 1477.000 1481.000 1492.000 1499.000 1512.000 1518.000 1523.000 1524.000 1552.000 1553.000 1554.000 1573.000 1576.000 1578.000 1582.000 1608.000 1611.000 1613.000 1616.000 1618.000 1627.000 1645.000 1651.000 1652.000 1653.000 1678.000 1680.000 1681.000 1685.000 1688.000 1690.000 1701.000 1702.000 1712.000 1730.000 1734.000 1740.000 1742.000 1754.000 1756.000 1757.000 1758.000 1760.000 1761.000 1768.000 1769.000 1770.000 1777.000 1782.000 1785.000 1786.000 1794.000 1802.000 1803.000 1808.000 1809.000 1817.000 1818.000 1819.000 1841.000 1850.000 1857.000 1891.000 1892.000 1897.000 1902.000 1910.000 1939.000 1950.000 1951.000 1975.000 2023.000 2024.000 2028.000 2045.000 2046.000 2049.000 2050.000 2053.000 2066.000 2068.000 2069.000 2077.000 2078.000 2079.000 2089.000 2099.000 2121.000 2126.000 2128.000 2132.000 2133.000 2137.000 2140.000 2141.000 2144.000 2145.000 2148.000 2151.000 2163.000 2167.000 2168.000 2180.000 2181.000 2185.000 2186.000 2193.000 2202.000 2205.000 2207.000 2211.000 2212.000 2216.000 2218.000 2224.000 2226.000 2230.000 2233.000 2234.000 2239.000 2240.000 2241.000 2244.000 2248.000 2256.000 2263.000 2267.000 2269.000 2270.000 2271.000 2272.000 2277.000 2293.000 2294.000 2296.000 2300.000 2301.000 2326.000 2329.000 2338.000 2364.000 2408.000 2409.000 2411.000 2428.000 2429.000 2430.000 2436.000 2448.000 2449.000 2454.000 2457.000 2458.000 2462.000 2472.000 2491.000 2516.000 2519.000 2522.000 2526.000 2531.000 2533.000 2539.000 2544.000 2547.000 2552.000 2563.000 2564.000 2587.000 2610.000 2622.000 2624.000 2635.000 2668.000 2672.000 2681.000 2688.000 2692.000 2697.000 2713.000 2714.000 2718.000 2719.000 2720.000 2729.000 2734.000 2735.000 2743.000 2744.000 2747.000 2748.000 2755.000 2756.000 2765.000 2770.000 2773.000 2776.000 2780.000 2781.000 2807.000 2815.000 2816.000 2827.000 2836.000 2840.000 2850.000 2851.000 2862.000 2880.000 2881.000 2905.000 2928.000 2940.000 2943.000 2944.000 2946.000 2958.000 2965.000 2979.000 2980.000 2999.000 3002.000 3005.000 3026.000 3053.000 3056.000 3059.000 3062.000 3067.000 3105.000 3122.000 3127.000 3161.000 3162.000 3165.000 3172.000 3218.000 3222.000 3227.000 3267.000 3269.000 3306.000 3316.000 3358.000 3382.000 3385.000 3419.000 3425.000 3442.000 3450.000 3451.000 3460.000 3463.000 3472.000 3475.000 3487.000 3497.000 3518.000 3521.000 3523.000 3532.000 3535.000 3537.000 3544.000 3547.000 3552.000 3554.000 3557.000 3561.000 3582.000 3594.000 3599.000 3600.000 3644.000 3645.000 3658.000 3705.000 3716.000 3717.000 3718.000 3729.000 3730.000 3737.000 3744.000 3751.000 3757.000 3771.000 3774.000 3779.000 3781.000 3783.000 3797.000 3809.000 3824.000 3825.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

		3833.000 3835.000 3837.000 3841.000 3853.000 3879.000 3891.000 3898.000 3899.000 3904.000
		3926.000 3928.000 3963.000 3967.000 3969.000 3970.000 3975.000 4001.000 4012.000 4019.000
		4030.000 4037.000 4038.000 4062.000 4080.000 4103.000 4138.000 4142.000 4145.000 4186.000
		4199.000 4216.000 4219.000 4228.000 4233.000 4242.000 4244.000 4248.000 4257.000 4264.000
		4267.000 4268.000 4286.000 4287.000 4300.000 4304.000 4376.000 4409.000 4410.000 4412.000
		4424.000 4435.000 4436.000 4457.000 4464.000 4469.000 4473.000 4481.000 4487.000 4500.000
		4501.000 4511.000 4525.000 4526.000 4534.000 4540.000 4541.000 4545.000 4546.000 4550.000
		4558.000 4561.000 4562.000 4566.000 4568.000 4569.000 4570.000 4573.000 4574.000 4577.000
		4583.000 4585.000 4586.000 4588.000 4594.000 4596.000 4598.000 4599.000 4600.000 4604.000
		4606.000 4613.000 4630.000 4632.000 4636.000 4638.000 4639.000 4649.000 4659.000 4670.000
		4682.000 4695.000 4704.000 4710.000 4723.000 4729.000 4741.000 4750.000 4760.000 4762.000
		4763.000 4785.000 4790.000 4817.000 4826.000 4834.000 4844.000 4848.000 4850.000 4851.000
		4869.000 4871.000 4873.000 4879.000 4881.000 4882.000 4892.000 4896.000 4907.000 4918.000
		4923.000 4936.000 4943.000 4948.000 4962.000 4964.000 4970.000 4971.000 4991.000 4993.000
		4999.000 5000.000 5013.000 5023.000 5061.000 5068.000 5077.000 5078.000 5089.000 5090.000
		5092.000 5095.000 5097.000 5099.000 5112.000 5115.000 5124.000 5130.000 5132.000 5135.000
		5136.000 5137.000 5140.000 5142.000 5154.000 5159.000 5160.000 5162.000 5170.000 5174.000
		5175.000 5177.000 5185.000 5186.000 5188.000 5189.000 5190.000 5192.000 5193.000 5197.000
		5198.000 5200.000 5212.000 5216.000 5223.000 5233.000 5238.000 5243.000 5247.000 5251.000
		5252.000 5253.000 5257.000 5258.000 5274.000 5280.000 5297.000 5301.000 5302.000 5303.000
		5309.000 5313.000 5316.000 5326.000 5328.000 5330.000 5343.000 5345.000 5347.000 5353.000
		5354.000 5355.000 5358.000 5370.000 5389.000 5402.000 5411.000 5431.000 5432.000 5434.000
		5435.000 5437.000 5450.000 5452.000 5456.000 5457.000 5458.000 5464.000 5466.000 5504.000
		5510.000 5531.000 5535.000 5541.000 5543.000 5544.000 5547.000 5566.000 5568.000 5571.000
		5592.000 5618.000 5728.000 5729.000 5735.000 5738.000 5749.000 5760.000 5799.000 5802.000
		5804.000 5812.000 5814.000 5815.000 5816.000 5817.000 5822.000 5851.000 5886.000 5888.000
		5898.000 5899.000 5903.000 5905.000 5912.000 5926.000 5934.000 5946.000 5947.000 5956.000
		5973.000 5984.000 5987.000 5992.000 5996.000 6006.000 6010.000 6017.000 6028.000 6035.000
		6036.000 6039.000 6041.000 6045.000 6053.000 6068.000 6077.000 6088.000 6092.000 6097.000
		6102.000 6108.000 6118.000 6121.000 6128.000 6130.000 6147.000 6153.000 6162.000 6195.000
		6206.000 6210.000 6214.000 6230.000 6238.000 6260.000 6269.000 6270.000 6292.000 6307.000
		6308.000 6317.000 6328.000 6339.000 6340.000 6383.000 6385.000 6386.000 6394.000 6396.000
		6399.000 6401.000 6405.000 6407.000 6408.000 6412.000 6417.000 6418.000 6420.000 6425.000
		6441.000 6450.000 6471.000 6497.000 6501.000 6515.000 6535.000 6543.000 6545.000 6567.000
		6583.000 6587.000 6595.000 6643.000 6658.000 6664.000 6692.000 6694.000 6719.000 6729.000
		6730.000 6732.000 6740.000 6745.000 6754.000 6755.000 6805.000 6806.000 6815.000 6816.000
		6867.000 6868.000 6878.000 6880.000 6935.000 6937.000 6992.000 6993.000 7001.000 7048.000
		7049.000 7065.000 7090.000 7104.000 7136.000 7139.000 7157.000 7201.000 7203.000 7205.000
		7234.000 7273.000 7277.000 7341.000 7344.000 7348.000 7357.000 7426.000 7430.000 7440.000
		7453.000 7454.000 7461.000 7477.000 7481.000 7486.000 7492.000 7494.000 7500.000 7505.000
		7507.000 7515.000 7517.000 7524.000 7529.000 7533.000 7539.000 7549.000 7553.000 7558.000
		7564.000 7567.000 7573.000 7577.000 7580.000 7590.000 7592.000 7599.000 7604.000 7609.000
		7613.000 7618.000 7619.000 7620.000 7639.000 7655.000 7661.000 7662.000 7664.000 7673.000
		7685.000 7696.000 7702.000 7704.000 7716.000 7728.000 7731.000 7740.000 7748.000 7749.000
		7809.000 7818.000 7827.000 7828.000 7834.000 7836.000 7842.000 7856.000 7865.000 7866.000
		7877.000 7880.000 7891.000 7896.000 7898.000 7908.000 7927.000 7937.000 7950.000 7953.000
		7955.000 7958.000 7959.000 7960.000 7961.000 7962.000 7985.000 7989.000 7991.000 8010.000
		8016.000 8019.000 8020.000 8023.000 8034.000 8038.000 8053.000 8054.000 8055.000 8059.000
IGNSTOP	STRING	REFERENCED AT : 1087.000 1510.000 1617.000 5168.000 5192.000 5892.000 5895.000 7839.000
IMMED	LABEL	REFERENCED AT : 4550.000 8144.000
INCREMENT.ACTIVE.COUNT	LABEL	REFERENCED AT : 7541.000 7636.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

INCRN	STRING	REFERENCED AT : 2129.000 2138.000 2149.000 2208.000 2219.000 3626.000 3627.000 3628.000 3629.000 3630.000 3631.000 3705.000 3711.000 3720.000 3929.000 4030.000 4122.000 4133.000 4267.000 4268.000 7057.000 7149.000
INDIR	STRING	REFERENCED AT : 2526.000 2533.000 3472.000 3518.000 3535.000 4632.000 4638.000 4844.000 4850.000 4871.000 4881.000 4964.000 4970.000 4993.000 4999.000 5092.000 5112.000 5130.000 5135.000 5189.000 5212.000 5297.000 5303.000 5452.000 5458.000 5568.000 6730.000 6754.000 6815.000 6878.000 6935.000 6992.000 7048.000 7090.000 7136.000 8053.000
INITIALIZE.55.EXIT	LABEL	REFERENCED AT : 1785.000 1786.000 1789.000
INITIALIZE.75.EXIT	LABEL	REFERENCED AT : 1782.000 1787.000
INITIALIZE.CPU.MODE	LABEL	REFERENCED AT : 1237.000 1769.000 1770.000 1776.000
INITIALIZE.LOAD.MAP	LABEL	REFERENCED AT : 5356.000 6463.000
INST.NPM.FLG	VARIABLE	REFERENCED AT : 539.000 3115.000 3117.000
INST.PE	LABEL	REFERENCED AT : 713.000 3096.000
INST.PE.FLG	VARIABLE	REFERENCED AT : 537.000 3098.000
INSTMIUER	STRING	REFERENCED AT : 712.000
INSTNORESP	STRING	REFERENCED AT : 718.000
INSTR.SKIP	LABEL	REFERENCED AT : 1751.000 3154.000
INSTRUCTION.NONPRESENT	LABEL	REFERENCED AT : 720.000 3114.000
INSTRUCTION.TIMEOUT	LABEL	REFERENCED AT : 745.000 3116.000
INSTTIMEOUT	STRING	REFERENCED AT : 743.000
INT.DRT.TIMEOUT	LABEL	REFERENCED AT : 874.000 3030.000 3032.000 6397.000 6413.000
INT.ERR.EXIT1	LABEL	REFERENCED AT : 3000.000 3003.000 3006.000 3009.000 3027.000 3034.000
INT.ERR.EXIT2	LABEL	REFERENCED AT : 3010.000 3356.000
INT.ERR.FLG	VARIABLE	REFERENCED AT : 528.000 3000.000 3000.000 3003.000 3003.000 3006.000 3008.000 3025.000 3030.000 3033.000 7470.000
INT.EXIT	LABEL	REFERENCED AT : 7463.000
INT.IO.ERR1	LABEL	REFERENCED AT : 833.000 2996.000 6403.000
INT.IO.ERR2	LABEL	REFERENCED AT : 854.000 3024.000 3355.000 6422.000
INT.IO.ERR3	LABEL	REFERENCED AT : 2998.000 3026.000
INT.RDY.TIMEOUT	LABEL	REFERENCED AT : 842.000 3029.000 6410.000
INTERRUPT	LABEL	REFERENCED AT : 788.000 799.000 3174.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
INTERRUPT.ERROR          LABEL    REFERENCED AT :   847.000  853.000  931.000

INTERRUPT.EXIT           LABEL    REFERENCED AT :   918.000  924.000  971.000

INTLVL                   REGISTER REFERENCED AT :   787.000 1840.000 2244.000 2566.000 3164.000 5233.000 6741.000 7716.000
                                  7725.000

INTR                     VARIABLE REFERENCED AT :   266.000  801.000  858.000 1124.000 1746.000 1894.000 1928.000 2536.000
                                  2573.000 2584.000 2593.000 2605.000 2618.000 2678.000 2710.000 2713.000 3152.000 3155.000
                                  3162.000 3172.000 5240.000 5245.000

INTR.CTL.ERR             LABEL    REFERENCED AT :  2187.000

INTR.CTL.EXIT            LABEL    REFERENCED AT :  2612.000 2715.000

INTR.CTL.IO              LABEL    REFERENCED AT :  4451.000 8159.000

INTR.CTL.RTN             LABEL    REFERENCED AT :  4455.000 6870.000

INTR.ERR1                LABEL    REFERENCED AT :   832.000  835.000

INTR.ERR2                LABEL    REFERENCED AT :   841.000

INTR.ERR3                LABEL    REFERENCED AT :   945.000

INTR.ERR4                LABEL    REFERENCED AT :   873.000

INTR.ERR6                LABEL    REFERENCED AT :  6402.000 6405.000

INTR.ERR7                LABEL    REFERENCED AT :  6409.000

INTR.ERR8                LABEL    REFERENCED AT :  6418.000 6421.000

INTR.ERR9                LABEL    REFERENCED AT :  3228.000

INTR.INC                 LABEL    REFERENCED AT :  2207.000 2210.000

INTR.IOM.TYPE            LABEL    REFERENCED AT :  2668.000 2683.000

INTR.LIMIT.2.11          LABEL    REFERENCED AT :  2573.000 2584.000 2593.000 2605.000 2618.000 2708.000

INTR.RTOM.TYPE           LABEL    REFERENCED AT :  2671.000

INTR.WAIT.READY          LABEL    REFERENCED AT :   826.000  834.000

INTRENA                  STRING   REFERENCED AT :  1085.000 2519.000 4436.000

INTRLOC                  VARIABLE REFERENCED AT :   284.000  913.000 1291.000 1896.000 2248.000 2660.000 3323.000 7352.000
                                  7462.000 7483.000 7555.000 7627.000

INTRTAB                  VARIABLE REFERENCED AT :   272.000  818.000  820.000  859.000  908.000  909.000  927.000  927.000
                                   934.000  998.000  998.000 1292.000 1807.000 1898.000 2065.000 2076.000 2088.000 2088.000
                                  2180.000 2249.000 2255.000 2400.000 2400.000 2538.000 2536.000 2576.000 2576.000 2587.000
                                  2587.000 2597.000 2597.000 2609.000 2609.000 2621.000 2621.000 2662.000 5246.000 5259.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

			5259.000 6389.000 7255.000 7255.000 7273.000 7295.000 7298.000 7301.000 7307.000 7310.000 7316.000 7316.000 7324.000 7324.000 7327.000 7327.000 7330.000 7330.000 7335.000 7335.000 7339.000 7347.000 7351.000 7425.000 7425.000 7460.000 7461.000 7475.000 7481.000 7547.000 7553.000 7626.000 7651.000 7651.000
INVALID.IOM	LABEL	REFERENCED AT :	1897.000 2673.000 2692.000 2698.000
IO.REG.LOAD	LABEL	REFERENCED AT :	1889.000 2260.000 2726.000 7210.000
IO.REG.LOAD.EXIT	LABEL	REFERENCED AT :	1906.000 1909.000
IO.SEQ	LABEL	REFERENCED AT :	1314.000 1716.000 2066.000 2070.000 2090.000 2116.000
IO.TIM.FLG	VARIABLE	REFERENCED AT :	534.000 3003.000 3033.000 7441.000 7534.000 7605.000
IO.WDOT	LABEL	REFERENCED AT :	1821.000 2146.000 2150.000
IOCD.ADDR	VARIABLE	REFERENCED AT :	202.000 1908.000 1936.000 2289.000 2364.000 2381.000 2402.000 7239.000 7270.000
IOCD1	VARIABLE	REFERENCED AT :	230.000 2308.000 2310.000 2317.000 2331.000 2370.000 2379.000
IOCHBUSY	STRING	REFERENCED AT :	826.000 846.000 1311.000 1315.000 1802.000 2126.000 2141.000 2151.000 2186.000 2207.000 2218.000 2449.000 2458.000 2781.000 3005.000 3026.000 6399.000 6417.000 7492.000 7524.000 7564.000 7590.000
IONORESP	STRING	REFERENCED AT :	835.000 846.000 1276.000 1316.000 1518.000 1627.000 1645.000 1702.000 1802.000 2126.000 2141.000 2151.000 2205.000 2216.000 2277.000 2409.000 2428.000 2999.000 6405.000 6417.000 7277.000 7500.000 7573.000 7655.000 7673.000 7950.000
IORESPRDY	STRING	REFERENCED AT :	821.000 839.000 2077.000 2078.000 2132.000 2133.000 2211.000 2212.000 2429.000 2979.000 6407.000 7486.000 7505.000 7558.000 7577.000 7661.000
IORESPRDY.PENDING	LABEL	REFERENCED AT :	2134.000 2159.000 2213.000 2430.000
IORETRY	STRING	REFERENCED AT :	826.000 846.000 847.000 1311.000 1315.000 1802.000 1803.000 2128.000 2137.000 2148.000 2207.000 2218.000 6399.000 6417.000 6418.000 7492.000 7515.000 7564.000 7590.000
IOTIMEOUT	STRING	REFERENCED AT :	835.000 872.000 1276.000 1316.000 1518.000 1627.000 1645.000 1702.000 1802.000 2126.000 2141.000 2151.000 2205.000 2216.000 2234.000 2277.000 2409.000 2428.000 3002.000 6396.000 6405.000 6412.000 7440.000 7533.000 7604.000 7950.000
IPL	LABEL	REFERENCED AT :	1010.000 1267.000
IPL.55	LABEL	REFERENCED AT :	1771.000
IPL.AF	LABEL	REFERENCED AT :	1448.000 1452.000
IPL.CLEAR	LABEL	REFERENCED AT :	1377.000 1401.000 1463.000
IPL.CONTINUE	LABEL	REFERENCED AT :	1316.000 1321.000
IPL.CPU.MODE	LABEL	REFERENCED AT :	1285.000 1766.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

IPL.DELIMITER	LABEL	REFERENCED AT :	1304.000 1440.000 1441.000
IPL.DEV	LABEL	REFERENCED AT :	1308.000 1311.000 1325.000 1339.000 1341.000
IPL.DEV.LOAD	LABEL	REFERENCED AT :	1393.000 1399.000
IPL.EARLY.INT.ERR	LABEL	REFERENCED AT :	7703.000 7760.000
IPL.EDEV	LABEL	REFERENCED AT :	1370.000 1376.000
IPL.END	LABEL	REFERENCED AT :	1367.000 1373.000
IPL.ERR.FLG	VARIABLE	REFERENCED AT :	551.000 1319.000 7758.000
IPL.FLAGS	VARIABLE	REFERENCED AT :	170.000 1346.000 1352.000 1368.000 1370.000 1471.000 1471.000
IPL.FLD1	LABEL	REFERENCED AT :	1306.000 1460.000
IPL.FOLD	LABEL	REFERENCED AT :	1451.000 1455.000
IPL.HALT	LABEL	REFERENCED AT :	1307.000 1312.000 1318.000 1444.000 1445.000 1449.000
IPL.INTR	LABEL	REFERENCED AT :	1368.000 1400.000
IPL.MACRO.START	LABEL	REFERENCED AT :	1326.000 1349.000 1364.000 1372.000
IPL.NEXT	LABEL	REFERENCED AT :	1303.000 1331.000 1359.000 1375.000 1392.000 1395.000 1408.000
IPL.NEXT.BYTE	LABEL	REFERENCED AT :	1432.000 1462.000
IPL.NEXT.WORD	LABEL	REFERENCED AT :	1428.000 1461.000
IPL.READ.ENTRY	LABEL	REFERENCED AT :	1378.000 1402.000 1423.000
IPL.SCAN	LABEL	REFERENCED AT :	1346.000 1360.000
IPLSW	STRING	REFERENCED AT :	1009.000
IPU	STRING	REFERENCED AT :	979.000 1109.000 1492.000 1499.000 1782.000 1794.000 1850.000 1857.000 2099.000 5238.000 6441.000 6450.000 6692.000 6740.000 6806.000 6867.000 7865.000 7880.000 7891.000 7898.000 7937.000
IPU.BRI	LABEL	REFERENCED AT :	2512.000 6738.000
IPU.CALM.PROC	LABEL	REFERENCED AT :	5238.000 7906.000
IPU.CHECK	LABEL	REFERENCED AT :	7689.000
IPU.CTL	LABEL	REFERENCED AT :	4452.000 6865.000
IPU.ERROR.TRAP	LABEL	REFERENCED AT :	1797.000 1850.000 1861.000 6703.000 7872.000 7929.000
IPU.HALT	STRING	REFERENCED AT :	921.000 1496.000 7858.000 7927.000 7928.000 7930.000
IPU.POWER.DWN	LABEL	REFERENCED AT :	1109.000 6700.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

IPU.POWER.ON	LABEL	REFERENCED AT :	1116.000 6691.000
IPU.PWR.OUT	LABEL	REFERENCED AT :	1123.000 2098.000
IPU.START	STRING	REFERENCED AT :	748.000 777.000 1497.000 1498.000 7877.000
IPU.TEST.75	LABEL	REFERENCED AT :	7891.000 8012.000
IPU.TRAP	LABEL	REFERENCED AT :	977.000 7886.000
IPU.TRAP.CHECK	LABEL	REFERENCED AT :	3300.000 6439.000
IPU.TRAP.FLAG.TEST	LABEL	REFERENCED AT :	6441.000 7925.000
IPU.WAIT	LABEL	REFERENCED AT :	5425.000 6803.000
ISSUE.ICT	LABEL	REFERENCED AT :	930.000 2202.000 2212.000 2214.000 2220.000
ISSUE.INTR.CTL.SEQ	LABEL	REFERENCED AT :	2181.000 2185.000 2190.000
JAV1	LABEL	REFERENCED AT :	6758.000 6760.000 6764.000 6780.000
JAV1.01	LABEL	REFERENCED AT :	6760.000
JAV1.03	LABEL	REFERENCED AT :	6764.000
JAV1.13	LABEL	REFERENCED AT :	6780.000
JAV3	LABEL	REFERENCED AT :	6819.000 6821.000 6825.000 6841.000
JAV3.01	LABEL	REFERENCED AT :	6821.000
JAV3.03	LABEL	REFERENCED AT :	6825.000
JAV3.13	LABEL	REFERENCED AT :	6841.000
JAV5	LABEL	REFERENCED AT :	6884.000 6886.000 6890.000 6909.000
JAV5.01	LABEL	REFERENCED AT :	6886.000
JAV5.03	LABEL	REFERENCED AT :	6890.000
JAV5.13	LABEL	REFERENCED AT :	6909.000
JAV7	LABEL	REFERENCED AT :	6941.000 6943.000 6947.000 6966.000
JAV7.01	LABEL	REFERENCED AT :	6943.000
JAV7.03	LABEL	REFERENCED AT :	6947.000
JAV7.13	LABEL	REFERENCED AT :	6966.000
JDV01	LABEL	REFERENCED AT :	7056.000 7058.000 7060.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

JDV01.01	LABEL	REFERENCED AT : 7058.000
JDV01.01.EXIT	LABEL	REFERENCED AT : 7059.000 7082.000
JDV01.03	LABEL	REFERENCED AT : 7060.000
JDV02	LABEL	REFERENCED AT : 7148.000 7150.000 7152.000
JDV02.01	LABEL	REFERENCED AT : 7150.000
JDV02.01.EXIT	LABEL	REFERENCED AT : 7151.000 7182.000
JDV02.02	LABEL	REFERENCED AT : 7152.000
JMP.TBL	LABEL	REFERENCED AT : 7246.000 7287.000 7291.000 7293.000 7296.000 7299.000 7302.000 7305.000 7308.000 7311.000 7315.000 7318.000 7320.000 7323.000 7326.000 7329.000 7332.000
JNV1	LABEL	REFERENCED AT : 6766.000 6769.000 6772.000 6775.000
JNV1.01	LABEL	REFERENCED AT : 6769.000
JNV1.03	LABEL	REFERENCED AT : 6772.000
JNV1.05	LABEL	REFERENCED AT : 6763.000 6775.000
JNV10	LABEL	REFERENCED AT : 7069.000 7071.000 7073.000 7075.000
JNV10.01	LABEL	REFERENCED AT : 7071.000
JNV10.01EXIT	LABEL	REFERENCED AT : 7072.000 7079.000
JNV10.03	LABEL	REFERENCED AT : 7073.000
JNV10.05	LABEL	REFERENCED AT : 7075.000
JNV11	LABEL	REFERENCED AT : 7113.000 7115.000 7119.000 7123.000
JNV11.01	LABEL	REFERENCED AT : 7115.000
JNV11.03	LABEL	REFERENCED AT : 7119.000
JNV11.05	LABEL	REFERENCED AT : 7123.000
JNV12	LABEL	REFERENCED AT : 7161.000 7163.000 7167.000 7171.000
JNV12.01	LABEL	REFERENCED AT : 7163.000
JNV12.01EXIT	LABEL	REFERENCED AT : 7165.000 7178.000
JNV12.03	LABEL	REFERENCED AT : 7167.000
JNV12.05	LABEL	REFERENCED AT : 7171.000
JNV2	LABEL	REFERENCED AT : 6784.000 6786.000 6789.000 6792.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

JNV2.01	LABEL	REFERENCED AT :	6786.000
JNV2.03	LABEL	REFERENCED AT :	6789.000
JNV2.05	LABEL	REFERENCED AT :	6792.000
JNV3	LABEL	REFERENCED AT :	6827.000 6830.000 6833.000 6836.000
JNV3.01	LABEL	REFERENCED AT :	6830.000
JNV3.03	LABEL	REFERENCED AT :	6833.000
JNV3.05	LABEL	REFERENCED AT :	6824.000 6836.000
JNV4	LABEL	REFERENCED AT :	6845.000 6847.000 6850.000 6853.000
JNV4.01	LABEL	REFERENCED AT :	6847.000
JNV4.03	LABEL	REFERENCED AT :	6850.000
JNV4.05	LABEL	REFERENCED AT :	6853.000
JNV5	LABEL	REFERENCED AT :	6892.000 6895.000 6899.000 6903.000
JNV5.01	LABEL	REFERENCED AT :	6895.000
JNV5.03	LABEL	REFERENCED AT :	6899.000
JNV5.05	LABEL	REFERENCED AT :	6889.000 6903.000
JNV6	LABEL	REFERENCED AT :	6912.000 6914.000 6918.000 6922.000
JNV6.01	LABEL	REFERENCED AT :	6914.000
JNV6.03	LABEL	REFERENCED AT :	6918.000
JNV6.05	LABEL	REFERENCED AT :	6922.000
JNV7	LABEL	REFERENCED AT :	6949.000 6952.000 6956.000 6960.000
JNV7.01	LABEL	REFERENCED AT :	6952.000
JNV7.03	LABEL	REFERENCED AT :	6956.000
JNV7.05	LABEL	REFERENCED AT :	6946.000 6960.000
JNV8	LABEL	REFERENCED AT :	6970.000 6972.000 6976.000 6980.000
JNV8.01	LABEL	REFERENCED AT :	6972.000
JNV8.03	LABEL	REFERENCED AT :	6976.000
JNV8.05	LABEL	REFERENCED AT :	6980.000
JNV9	LABEL	REFERENCED AT :	7007.000 7009.000 7012.000 7015.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

JNV9.01	LABEL	REFERENCED AT : 7009.000
JNV9.03	LABEL	REFERENCED AT : 7012.000
JNV9.05	LABEL	REFERENCED AT : 7015.000
JUMP.WCS	LABEL	REFERENCED AT : 5568.000 5573.000
JUMPD	STRING	REFERENCED AT : 639.000 683.000 1092.000 1097.000 3443.000 3518.000 3731.000 4459.000 4464.000 4501.000 4511.000 4526.000 4534.000 4541.000 4546.000 4550.000 4561.000 4563.000 4569.000 4571.000 4574.000 4632.000 4638.000 4871.000 4881.000 4964.000 4970.000 5013.000 5061.000 5099.000 5125.000 5142.000 5176.000 5178.000 5199.000 5202.000 5317.000 5330.000 6719.000 6732.000 6734.000 7028.000 7038.000 7955.000
JUMPJ	STRING	REFERENCED AT : 685.000 996.000 1000.000 1078.000 1138.000 1261.000 1324.000 1439.000 1473.000 1482.000 1484.000 1499.000 1578.000 1618.000 1701.000 1710.000 1712.000 1730.000 1740.000 1743.000 1754.000 1763.000 1765.000 1788.000 1790.000 1802.000 1910.000 1939.000 1977.000 2024.000 2058.000 2069.000 2077.000 2145.000 2167.000 2234.000 2244.000 2248.000 2249.000 2640.000 2681.000 2697.000 2701.000 2713.000 2714.000 2780.000 2782.000 2905.000 2906.000 2931.000 2949.000 2965.000 2981.000 3161.000 3162.000 3172.000 3246.000 3269.000 3274.000 3382.000 3390.000 3572.000 3574.000 3576.000 3835.000 3838.000 3988.000 4012.000 4013.000 4113.000 4278.000 4280.000 4298.000 4302.000 4306.000 4308.000 4376.000 4380.000 4406.000 4410.000 4504.000 4817.000 4826.000 4834.000 5117.000 5219.000 5223.000 5343.000 5368.000 5547.000 5592.000 5595.000 5614.000 5622.000 5789.000 5794.000 5807.000 5828.000 5899.000 5916.000 5920.000 5942.000 6059.000 6063.000 6085.000 6225.000 6302.000 6386.000 6425.000 6429.000 6501.000 6515.000 6550.000 6567.000 6609.000 6638.000 6645.000 6745.000 7280.000 7341.000 7414.000 7428.000 7442.000 7468.000 7481.000 7498.000 7521.000 7539.000 7543.000 7553.000 7571.000 7584.000 7607.000 7630.000 7641.000 7675.000 7677.000 7685.000 7786.000 7801.000 7828.000 7958.000 7962.000 7963.000 7965.000 7967.000 7969.000 8043.000 8061.000
JUMP7	STRING	REFERENCED AT : 1091.000 1095.000 1678.000 1761.000 1823.000 2279.000 2413.000 2501.000 2716.000 3479.000 3649.000 4500.000 4525.000 4540.000 4545.000 4553.000 4555.000 4573.000 4657.000 4661.000 4674.000 4687.000 4708.000 4713.000 4715.000 4727.000 4731.000 4733.000 4748.000 4796.000 4899.000 4928.000 4979.000 5004.000 5028.000 5040.000 5070.000 5096.000 5097.000 5138.000 5140.000 5147.000 5163.000 5175.000 5177.000 5198.000 5200.000 5253.000 5280.000 5328.000 5472.000 5578.000 6771.000 6774.000 6779.000 6788.000 6791.000 6796.000 6832.000 6835.000 6840.000 6849.000 6852.000 6857.000 6898.000 6902.000 6908.000 6917.000 6921.000 6927.000 6955.000 6959.000 6965.000 6975.000 6979.000 6985.000 7011.000 7014.000 7019.000 7074.000 7078.000 7081.000 7084.000 7118.000 7122.000 7128.000 7176.000 7181.000 7185.000 7355.000 7856.000 7908.000 7953.000
JWCS	LABEL	REFERENCED AT : 5565.000 8547.000
JWCS.1	LABEL	REFERENCED AT : 5567.000 5571.000
L00000XXX	LABEL	REFERENCED AT : 3691.000 3695.000
L0000XXXX	LABEL	REFERENCED AT : 3686.000 3696.000 3699.000
L000XXXXX	LABEL	REFERENCED AT : 3687.000 3702.000 3707.000
LA	LABEL	REFERENCED AT : 3440.000 8107.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

LA.EXIT	LABEL	REFERENCED AT : 3478.000
LA1	LABEL	REFERENCED AT : 3449.000 8257.000
LA2	LABEL	REFERENCED AT : 3451.000 3452.000 3470.000
LASI	LABEL	REFERENCED AT : 4552.000 8398.000 8399.000 8400.000
LATERR	STRING	REFERENCED AT : 671.000 4464.000 4511.000 4534.000 4550.000 5013.000
LATERR.TEST	LABEL	REFERENCED AT : 696.000 698.000 711.000
LATERRW	STRING	REFERENCED AT : 698.000 3442.000 3729.000 4558.000 4566.000 4583.000 4594.000 4630.000 4760.000 4869.000 4962.000 4991.000 5124.000 5154.000 5185.000 5274.000 5450.000 6719.000 6868.000
LATERR7	STRING	REFERENCED AT : 5068.000 5077.000 5090.000 5233.000
LCS	LABEL	REFERENCED AT : 4468.000 8378.000
LCS2	LABEL	REFERENCED AT : 4472.000
LD.PSEUDO.MAP	LABEL	REFERENCED AT : 6629.000 6634.000 6639.000
LDMAP	STRING	REFERENCED AT : 603.000 6546.000 6595.000 6643.000
LDSTOP	STRING	REFERENCED AT : 6244.000
LEA	LABEL	REFERENCED AT : 3441.000 8147.000
LEA.75	LABEL	REFERENCED AT : 3460.000 3462.000
LEA.EXIT	LABEL	REFERENCED AT : 3472.000 3475.000 3477.000
LEA.INDIRECT	LABEL	REFERENCED AT : 3473.000
LEA1	LABEL	REFERENCED AT : 3459.000 8318.000
LEA2	LABEL	REFERENCED AT : 3461.000 3463.000 3471.000
LEAP	STRING	REFERENCED AT : 2256.000 2263.000
LEAR	LABEL	REFERENCED AT : 5449.000 8126.000
LEAR.EXIT	LABEL	REFERENCED AT : 5460.000 5465.000
LEAR.INDIRECT	LABEL	REFERENCED AT : 5453.000 5458.000
LEAR.NOT.INDIRECT	LABEL	REFERENCED AT : 5452.000 5461.000
LEXTPROC	LABEL	REFERENCED AT : 2632.000 3442.000 3729.000 4453.000 4465.000 4512.000 4535.000 4551.000 4559.000 4567.000 4583.000 4594.000 4630.000 4760.000 4869.000 4962.000 4991.000 5014.000 5069.000 5079.000 5091.000 5154.000 5274.000 5450.000 6720.000 6869.000
LF	LABEL	REFERENCED AT : 4775.000 8317.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

LF.EXIT LABEL REFERENCED AT : 4794.000

LF1 LABEL REFERENCED AT : 4779.000 4783.000

LFC1V STRING REFERENCED AT : 94.000 6755.000 6993.000 7049.000 7139.000

LINK STRING REFERENCED AT :
651.000 660.000 676.000 713.000 715.000 717.000 720.000 740.000
742.000 745.000 747.000 750.000 768.000 771.000 773.000 776.000 802.000 803.000
822.000 825.000 827.000 845.000 848.000 856.000 860.000 884.000 897.000 921.000
926.000 930.000 935.000 969.000 997.000 1014.000 1020.000 1021.000 1022.000 1024.000
1040.000 1059.000 1072.000 1073.000 1074.000 1075.000 1197.000 1206.000 1209.000 1218.000
1220.000 1222.000 1226.000 1229.000 1235.000 1237.000 1268.000 1275.000 1285.000 1295.000
1309.000 1314.000 1327.000 1328.000 1329.000 1334.000 1336.000 1339.000 1341.000 1343.000
1354.000 1357.000 1377.000 1378.000 1387.000 1401.000 1402.000 1430.000 1458.000 1459.000
1466.000 1468.000 1470.000 1513.000 1516.000 1575.000 1593.000 1613.000 1620.000 1644.000
1656.000 1664.000 1706.000 1725.000 1739.000 1747.000 1759.000 1800.000 1801.000 1808.000
1813.000 1821.000 1842.000 1864.000 1906.000 1908.000 1933.000 1937.000 2051.000 2067.000
2124.000 2125.000 2129.000 2136.000 2138.000 2147.000 2149.000 2179.000 2183.000 2188.000
2191.000 2201.000 2204.000 2208.000 2215.000 2219.000 2243.000 2247.000 2260.000 2278.000
2351.000 2377.000 2381.000 2402.000 2422.000 2427.000 2432.000 2447.000 2456.000 2464.000
2470.000 2490.000 2494.000 2496.000 2530.000 2536.000 2546.000 2548.000 2573.000 2574.000
2584.000 2585.000 2593.000 2594.000 2605.000 2606.000 2618.000 2619.000 2624.000 2628.000
2633.000 2686.000 2690.000 2691.000 2726.000 2730.000 2731.000 2738.000 2739.000 2740.000
2772.000 2796.000 2804.000 2823.000 2833.000 2858.000 2869.000 2897.000 2915.000 3067.000
3068.000 3133.000 3137.000 3153.000 3154.000 3176.000 3178.000 3202.000 3204.000 3219.000
3244.000 3306.000 3347.000 3350.000 3353.000 3357.000 3389.000 3416.000 3426.000 3474.000
3504.000 3536.000 3560.000 3802.000 3810.000 3820.000 3822.000 3824.000 3827.000 3890.000
3901.000 3910.000 3913.000 3966.000 3968.000 3972.000 3980.000 3984.000 3991.000 4029.000
4078.000 4185.000 4187.000 4193.000 4198.000 4200.000 4206.000 4210.000 4215.000 4216.000
4222.000 4226.000 4228.000 4229.000 4231.000 4235.000 4240.000 4242.000 4251.000 4255.000
4257.000 4388.000 4430.000 4587.000 4655.000 4685.000 4725.000 4743.000 4873.000 4938.000
5052.000 5093.000 5113.000 5115.000 5131.000 5241.000 5254.000 5294.000 5321.000 5451.000
5455.000 5471.000 5499.000 5521.000 5569.000 5594.000 5598.000 5613.000 5725.000 5745.000
5751.000 5761.000 5797.000 5798.000 5800.000 5801.000 5803.000 5810.000 5831.000 5837.000
5847.000 5855.000 5865.000 5870.000 5875.000 5882.000 5883.000 5884.000 5885.000 5887.000
5889.000 5911.000 5918.000 5923.000 5924.000 5925.000 5931.000 5953.000 5958.000 5974.000
5977.000 5981.000 5993.000 5998.000 6001.000 6007.000 6011.000 6013.000 6025.000 6037.000
6038.000 6040.000 6044.000 6048.000 6055.000 6056.000 6061.000 6065.000 6069.000 6073.000
6080.000 6082.000 6089.000 6094.000 6105.000 6110.000 6115.000 6120.000 6125.000 6126.000
6127.000 6129.000 6134.000 6140.000 6150.000 6166.000 6169.000 6178.000 6192.000 6198.000
6205.000 6209.000 6224.000 6227.000 6231.000 6235.000 6239.000 6257.000 6261.000 6295.000
6298.000 6305.000 6310.000 6316.000 6319.000 6329.000 6332.000 6352.000 6356.000 6371.000
6398.000 6400.000 6416.000 6419.000 6431.000 6451.000 6465.000 6474.000 6480.000 6482.000
6506.000 6629.000 6634.000 6653.000 6881.000 6938.000 7096.000 7141.000 7210.000 7272.000
7339.000 7347.000 7453.000 7460.000 7479.000 7480.000 7487.000 7491.000 7514.000 7551.000
7552.000 7559.000 7563.000 7589.000 7654.000 7658.000 7672.000 7688.000 7692.000 7695.000
7700.000 7714.000 7730.000 7739.000 7743.000 7765.000 7766.000 7806.000 7807.000 7808.000
7810.000 7813.000 7855.000 7857.000 7880.000 7909.000 7911.000 7951.000 7959.000 7960.000
7961.000 7977.000 7979.000 8007.000 8028.000 8060.000

LM LABEL REFERENCED AT : 4694.000 8303.000

LM.DWORD LABEL REFERENCED AT : 4684.000 4695.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

LMAP	LABEL	REFERENCED AT :	2489.000 8428.000
LN	LABEL	REFERENCED AT :	4681.000 8304.000
LN.DWORD	LABEL	REFERENCED AT :	4683.000
LOAD.MAP	LABEL	REFERENCED AT :	6474.000 6482.000 6496.000 6516.000
LOAD.PSEUDO.MAP	LABEL	REFERENCED AT :	1235.000 6624.000
LOAD.WCS	LABEL	REFERENCED AT :	5598.000 5613.000 5616.000 5618.000
LOG.ADDR	VARIABLE	REFERENCED AT :	261.000
LOG.MAP	VARIABLE	REFERENCED AT :	267.000
LOOP.CLEAR.GPRS	LABEL	REFERENCED AT :	1153.000 1156.000
LOOP.CLEAR.MAP	LABEL	REFERENCED AT :	6542.000 6546.000
LOOP.EFFECTIVE.INDIR	LABEL	REFERENCED AT :	8047.000 8053.000
LOOP.LF	LABEL	REFERENCED AT :	4780.000 4790.000
LOOP.LOAD.MAP	LABEL	REFERENCED AT :	603.000 6584.000
LOOP.PSEUDO	LABEL	REFERENCED AT :	6641.000 6643.000
LOOP.STF	LABEL	REFERENCED AT :	4910.000 4923.000
LP.TD2	LABEL	REFERENCED AT :	2759.000
LP.TD4	LABEL	REFERENCED AT :	2756.000 2795.000
LP.TD8	LABEL	REFERENCED AT :	2755.000 2803.000
LPSD	LABEL	REFERENCED AT :	5292.000 8546.000
LPSD.ERR1	LABEL	REFERENCED AT :	5301.000 5310.000
LPSD.EXIT	LABEL	REFERENCED AT :	5318.000 8355.000
LPSD.FETCH	LABEL	REFERENCED AT :	5295.000 5300.000
LPSD.INDIRECT	LABEL	REFERENCED AT :	5298.000 5303.000
LPSD.RH	LABEL	REFERENCED AT :	5327.000
LPSD1	LABEL	REFERENCED AT :	5297.000 5304.000
LPSD2	LABEL	REFERENCED AT :	1844.000 5322.000 7753.000
LPSDCM	LABEL	REFERENCED AT :	5293.000 8549.000
LPSDCM.EXIT	LABEL	REFERENCED AT :	5320.000 8357.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
LXXXXXXXX                LABEL    REFERENCED AT :   3711.000 3721.000

LXXXXXXXX.A              LABEL    REFERENCED AT :   3684.000 3723.000

M.55.TRAP.HALT           LABEL    REFERENCED AT :   1320.000 3346.000 3358.000 7759.000 7822.000

M.75.TRAP.HALT           LABEL    REFERENCED AT :   3222.000 3299.000 3351.000

M.75.TRAP.RTN            LABEL    REFERENCED AT :   3302.000 6443.000

MACHINE.CHECK.STATUS     LABEL    REFERENCED AT :   3082.000 3177.000

MACHINE.CHECK.TRAP       LABEL    REFERENCED AT :   3175.000 8022.000

MAP.ADDR                 VARIABLE REFERENCED AT :    160.000 6467.000 6476.000 6484.000 6501.000 6516.000 6540.000

MAP.FAULT.TRAP           LABEL    REFERENCED AT :   3127.000 3131.000 7767.000

MAP.FAULT.TYPE           LABEL    REFERENCED AT :   1847.000 3143.000

MAP.INVALID.FLG          VARIABLE REFERENCED AT :    550.000 3132.000

MAP.MODE.INVALID.ERR     LABEL    REFERENCED AT :   2492.000 6659.000 7764.000

MAP.N.FOUND.FLG          VARIABLE REFERENCED AT :    544.000 6552.000

MAPADD                   LABEL    REFERENCED AT :   6657.000

MAPINVALID               STRING   REFERENCED AT :   2049.000 3122.000 3127.000 7827.000 7834.000

MAPMODE                  STRING   REFERENCED AT :   1477.000 2491.000 2498.000 3210.000 3384.000 3386.000 5352.000 5353.000
                                  5355.000 5358.000 6658.000 6660.000 6680.000 8023.000

MAR                      STRING   REFERENCED AT :   1036.000 1725.000 1837.000 1953.000 2040.000 2958.000 4475.000 4781.000
                                  4813.000 4822.000 4830.000 4911.000 4939.000 5026.000 5038.000 5077.000 5136.000 5170.000
                                  5306.000 5500.000 5508.000 5522.000 5524.000 5934.000 5977.000 6029.000 6664.000 6669.000
                                  6878.000 6935.000 7092.000 7137.000

MAR                      REGISTER REFERENCED AT :    639.000  674.000  682.000  864.000  896.000  942.000  950.000 1092.000
                                  1113.000 1118.000 1138.000 1185.000 1335.000 1356.000 1429.000 1572.000 1607.000 1725.000
                                  1726.000 1765.000 1812.000 1837.000 1929.000 1951.000 2396.000 2524.000 2532.000 2552.000
                                  2771.000 2774.000 2776.000 2777.000 2778.000 2806.000 2806.000 2835.000 2837.000 2839.000
                                  2841.000 2842.000 2872.000 3266.000 3271.000 3304.000 3306.000 3314.000 3318.000 3321.000
                                  3325.000 3349.000 3648.000 3716.000 3975.000 4141.000 4249.000 4265.000 4266.000 4768.000
                                  4776.000 4785.000 4813.000 4822.000 4830.000 4908.000 4918.000 4939.000 5069.000 5094.000
                                  5115.000 5136.000 5138.000 5140.000 5249.000 5250.000 5306.000 5330.000 5466.000 5508.000
                                  5524.000 5566.000 5571.000 5858.000 5977.000 5978.000 6092.000 6099.000 6117.000 6132.000
                                  6398.000 6499.000 6608.000 6642.000 6664.000 6666.000 6669.000 6671.000 6878.000 6935.000
                                  7092.000 7137.000 7381.000 7387.000 7396.000 7402.000 7408.000 7514.000 7589.000 7691.000
                                  7838.000 7955.000 8048.000

MARIX                    STRING   REFERENCED AT :   1575.000 2522.000 2531.000 3450.000 3460.000 3472.000 3474.000 3519.000
                                  3533.000 3537.000 4409.000 4630.000 4634.000 4846.000 4869.000 4873.000 4962.000 4966.000
                                  4991.000 4995.000 5013.000 5061.000 5089.000 5124.000 5132.000 5162.000 5186.000 5191.000
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

		5205.000 5222.000 5223.000 5299.000 5456.000 5458.000 5570.000 6719.000 6728.000 7169.000 8049.000
MASK	STRING	REFERENCED AT : 7005.000 7066.000 7110.000 7158.000
MEM.ERR.MSK	VARIABLE	REFERENCED AT : 3049.000 3054.000 3057.000 3060.000 3063.000
MEM.PE	LABEL	REFERENCED AT : 3095.000 3099.000
MEMORY.READ	LABEL	REFERENCED AT : 935.000 1343.000 1354.000 1357.000 1430.000 2014.000 2494.000 5569.000 7700.000 7714.000 7743.000
MEMORY.WRITE	LABEL	REFERENCED AT : 1334.000 1334.000 1336.000 1963.000 2377.000 2381.000 7688.000 7692.000
MERGE	LABEL	REFERENCED AT : 3622.000 3631.000 3632.000 3635.000 3641.000
MERGE.F.C.BITS	LABEL	REFERENCED AT : 8054.000 8055.000 8057.000
MID	VARIABLE	REFERENCED AT : 166.000 285.000 603.000 6595.000
MISC.INTERRUPTS	LABEL	REFERENCED AT : 678.000 691.000 1004.000 1064.000
MODE.HALT.CHECK	LABEL	REFERENCED AT : 7877.000 7935.000
MODE75	STRING	REFERENCED AT : 981.000 1685.000 1788.000 1790.000 2436.000 2688.000 3316.000 3358.000 3450.000 3460.000 3487.000 3497.000 4409.000 5343.000 7866.000 7958.000 7989.000 8014.000 8026.000
MODE75S	STRING	REFERENCED AT : 856.000 897.000 1613.000 1734.000 2564.000 2624.000 2718.000 3306.000 7201.000 7896.000 7959.000
MP.CORRECTION	LABEL	REFERENCED AT : 3824.000 3830.000
MP.CYCLES	LABEL	REFERENCED AT : 3822.000 4215.000 4226.000 4240.000 4255.000 4365.000 4366.000
MP.DOUBLE	LABEL	REFERENCED AT : 3659.000
MP.QUNEG	LABEL	REFERENCED AT : 3501.000 3825.000
MP.USEDI	LABEL	REFERENCED AT : 2971.000 3815.000
MPF	LABEL	REFERENCED AT : 3657.000 8361.000
MPFD	LABEL	REFERENCED AT : 3661.000 4182.000
MPFD.CRACK	LABEL	REFERENCED AT : 4187.000 4200.000 4275.000
MPFW	LABEL	REFERENCED AT : 3658.000 3884.000
MPI	LABEL	REFERENCED AT : 3742.000 8401.000
MPM	LABEL	REFERENCED AT : 3735.000 8314.000
MPR	LABEL	REFERENCED AT : 3793.000 8267.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

MPRDVR	LABEL	REFERENCED AT : 3728.000 8110.000 8111.000 8188.000 8189.000
MPROM	STRING	REFERENCED AT : 4368.000 4369.000 4370.000 4371.000 4372.000 4373.000 4374.000 4375.000 4380.000
MPY	STRING	REFERENCED AT : 4368.000 4369.000 4370.000 4371.000 4372.000 4373.000 4374.000 4375.000
MPY(	STRING	REFERENCED AT : 6995.000 6999.000 7001.000 7002.000 7057.000 7062.000 7098.000 7102.000 7104.000 7105.000 7149.000 7154.000
MPY.DVD.FLT.PT	LABEL	REFERENCED AT : 6717.000 8152.000
MPY.OFUF	LABEL	REFERENCED AT : 3544.000 3549.000
MPY.TAB	LABEL	REFERENCED AT : 3820.000 3890.000 4286.000 4288.000 4289.000 4290.000
MPYLOOP	STRING	REFERENCED AT : 7001.000 7104.000
MRLAS	LABEL	REFERENCED AT : 4648.000 8302.000 8305.000 8306.000
MRLAS.BYTE.WORD	LABEL	REFERENCED AT : 4659.000 4673.000
MRLAS.DWORD	LABEL	REFERENCED AT : 4651.000
MRLAS.HWORD	LABEL	REFERENCED AT : 4660.000
MRLAS1	LABEL	REFERENCED AT : 4649.000 4658.000 4682.000 4696.000
MRLOAD	LABEL	REFERENCED AT : 4629.000 8127.000 8128.000 8129.000 8130.000 8131.000 8137.000 8138.000 8139.000 8140.000 8141.000 8142.000 8143.000 8146.000 8153.000
MRLOAD.INDR	LABEL	REFERENCED AT : 4633.000 4639.000
MRLOG	LABEL	REFERENCED AT : 4669.000 8293.000 8294.000
MRLOG.B.H.W	LABEL	REFERENCED AT : 4672.000
MRLOG.DWORD	LABEL	REFERENCED AT : 4652.000 4670.000
MRSTORE	LABEL	REFERENCED AT : 4861.000 4868.000 8148.000 8150.000
MRSTORE.INDR	LABEL	REFERENCED AT : 4672.000 4862.000
MSCD	VARIABLE	REFERENCED AT : 157.000 6468.000 6483.000
MSD	VARIABLE	REFERENCED AT : 293.000 6505.000 6585.000 6589.000 6589.000
MSDLP	VARIABLE	REFERENCED AT : 233.000 6499.000 6609.000
MULTIPLY	LABEL	REFERENCED AT : 3797.000 3809.000 3818.000 3819.000
N	REGISTER	REFERENCED AT : 645.000 1156.000 1186.000 1259.000 2530.000 2888.000 2940.000 2943.000 2944.000 2946.000 2968.000 3560.000 3598.000 3847.000 3850.000 3852.000 3854.000 3891.000 3895.000 3916.000 3922.000 3923.000 3934.000 3968.000 4029.000 4031.000 4191.000 4195.000 4201.000 4202.000 4206.000 4616.000 4784.000 4914.000 5114.000 6150.000 6158.000 6159.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S F L 3 2 / 7 5 C P U

```
                                    6416.000 7280.000 7442.000

N.75.EXIT                   LABEL   REFERENCED AT :   2688.000 2694.000

N.COUNTER.ERROR.CODE        LABEL   REFERENCED AT :   7343.000 7348.000

N.EXT.IO.REG.LOAD.EXIT      LABEL   REFERENCED AT :   1933.000 1938.000

N.PSW2                      VARIABLE REFERENCED AT :    203.000  957.000 2495.000 2497.000 2693.000 3231.000 5349.000 5366.000
                                    6382.000 6465.000 6480.000 6548.000 7451.000

N.RESP.FLG                  VARIABLE REFERENCED AT :    530.000 3000.000 7279.000

NALU7                       STRING  REFERENCED AT :    696.000  697.000  700.000  824.000  910.000  954.000 1073.000 1075.000
                                    1346.000 1364.000 1573.000 1688.000 1740.000 1770.000 1841.000 2053.000 2121.000 2202.000
                                    2224.000 2226.000 2230.000 2672.000 2697.000 2944.000 2958.000 3162.000 3222.000 3451.000
                                    3463.000 3544.000 3718.000 4019.000 4103.000 4145.000 4219.000 4248.000 4264.000 4598.000
                                    4790.000 4923.000 5251.000 5316.000 5353.000 5464.000 5728.000 5888.000 5984.000 6017.000
                                    6026.000 6068.000 6108.000 6118.000 6130.000 6195.000 6208.000 6230.000 6238.000 6260.000
                                    6269.000 6292.000 6344.000 6535.000 7205.000 7430.000 7539.000 7639.000 7685.000 7728.000
                                    7748.000 7985.000 8019.000 8038.000

NCTR                        STRING  REFERENCED AT :   1089.000 1154.000 1456.000 1458.000 1532.000 1533.000 1594.000 1665.000
                                    4782.000 4912.000 5025.000 5040.000 6155.000 6157.000

NCTR0                       STRING  REFERENCED AT :   2301.000 3172.000 3382.000 3594.000 3716.000 3824.000 4138.000 4300.000
                                    4304.000 7344.000

NCTR4                       STRING  REFERENCED AT :   1156.000 1306.000 2024.000 2587.000

NCTR7                       STRING  REFERENCED AT :    603.000  828.000  840.000  849.000  931.000 1087.000 1113.000 1120.000
                                    1186.000 1259.000 1312.000 1392.000 1395.000 1576.000 1809.000 2069.000 2079.000 2185.000
                                    2244.000 2248.000 2408.000 2448.000 2457.000 2547.000 2552.000 2780.000 2980.000 3067.000
                                    3475.000 3835.000 3837.000 3963.000 3967.000 3969.000 3970.000 4216.000 4228.000 4242.000
                                    4257.000 4562.000 4569.000 4570.000 5160.000 5188.000 5192.000 5618.000 5812.000 5947.000
                                    6401.000 6408.000 6420.000 6501.000 6515.000 6543.000 6545.000 6583.000 6595.000 6643.000
                                    7001.000 7065.000 7104.000 7157.000 7273.000 7348.000 7461.000 7481.000 7494.000 7507.000
                                    7517.000 7553.000 7567.000 7580.000 7592.000 7662.000 7696.000 7731.000 7740.000 8055.000

NHMPREV                     STRING  REFERENCED AT :   4376.000

NL                          STRING  REFERENCED AT :   1390.000 1522.000 1593.000 1664.000 1807.000 2768.000 2887.000 3795.000
                                    4558.000 4566.000 5022.000 5036.000 6314.000 6535.000

NO.RETRY.NOT.BUSY           LABEL   REFERENCED AT :   7565.000 7572.000

NOD                         STRING  REFERENCED AT :    603.000  672.000  692.000  694.000  695.000  698.000  699.000  788.000
                                     812.000  816.000  818.000  820.000  858.000  860.000  865.000  908.000  932.000  947.000
                                     952.000  955.000  963.000  964.000  965.000  966.000 1002.000 1009.000 1011.000 1012.000
                                    1013.000 1016.000 1045.000 1046.000 1047.000 1072.000 1074.000 1085.000 1096.000 1097.000
                                    1156.000 1184.000 1186.000 1259.000 1275.000 1316.000 1322.000 1323.000 1344.000 1346.000
                                    1362.000 1365.000 1366.000 1367.000 1368.000 1370.000 1437.000 1438.000 1439.000 1442.000
                                    1443.000 1446.000 1447.000 1521.000 1550.000 1551.000 1610.000 1649.000 1650.000 1677.000
                                    1686.000 1716.000 1739.000 1752.000 1756.000 1768.000 1783.000 1815.000 1816.000 1817.000
                                    1839.000 1844.000 1900.000 2027.000 2049.000 2058.000 2065.000 2076.000 2077.000 2117.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

2141.000 2143.000 2145.000 2161.000 2179.000 2180.000 2204.000 2222.000 2223.000 2226.000
2228.000 2237.000 2238.000 2244.000 2255.000 2261.000 2265.000 2268.000 2270.000 2271.000
2291.000 2292.000 2293.000 2294.000 2300.000 2301.000 2327.000 2343.000 2362.000 2408.000
2411.000 2448.000 2460.000 2496.000 2529.000 2537.000 2540.000 2608.000 2620.000 2666.000
2691.000 2695.000 2711.000 2712.000 2718.000 2720.000 2727.000 2732.000 2734.000 2735.000
2741.000 2742.000 2743.000 2744.000 2745.000 2746.000 2747.000 2748.000 2755.000 2763.000
2772.000 2775.000 2805.000 2815.000 2825.000 2834.000 2836.000 2850.000 2860.000 2880.000
2881.000 2914.000 2926.000 2938.000 2940.000 2941.000 2942.000 2943.000 2944.000 2946.000
3107.000 3159.000 3160.000 3220.000 3527.000 3536.000 3542.000 3547.000 3548.000 3560.000
3598.000 3675.000 3716.000 3737.000 3751.000 3762.000 3769.000 3771.000 3773.000 3780.000
3785.000 3807.000 3822.000 3841.000 3849.000 3853.000 3879.000 3881.000 3897.000 3898.000
3901.000 3905.000 3912.000 3925.000 3926.000 3928.000 3977.000 4004.000 4011.000 4012.000
4035.000 4036.000 4060.000 4080.000 4084.000 4091.000 4092.000 4103.000 4141.000 4184.000
4217.000 4245.000 4262.000 4266.000 4271.000 4284.000 4285.000 4379.000 4380.000 4410.000
4442.000 4487.000 4520.000 4539.000 4544.000 4555.000 4584.000 4596.000 4597.000 4610.000
4727.000 4731.000 4733.000 4744.000 4748.000 4784.000 4914.000 4977.000 4982.000 5004.000
5027.000 5131.000 5134.000 5135.000 5219.000 5245.000 5249.000 5250.000 5251.000 5252.000
5257.000 5268.000 5271.000 5279.000 5314.000 5321.000 5324.000 5329.000 5343.000 5345.000
5346.000 5347.000 5351.000 5452.000 5457.000 5462.000 5464.000 5544.000 5545.000 5593.000
5594.000 5726.000 5748.000 5798.000 5801.000 5803.000 5812.000 5814.000 5815.000 5816.000
5887.000 5901.000 5903.000 5916.000 5954.000 5982.000 5985.000 5994.000 6003.000 6008.000
6015.000 6033.000 6038.000 6040.000 6051.000 6066.000 6075.000 6090.000 6095.000 6101.000
6116.000 6117.000 6151.000 6160.000 6193.000 6206.000 6228.000 6236.000 6258.000 6267.000
6290.000 6326.000 6383.000 6384.000 6385.000 6386.000 6389.000 6398.000 6424.000 6432.000
6500.000 6533.000 6546.000 6727.000 6730.000 6731.000 6757.000 6759.000 6762.000 6767.000
6771.000 6779.000 6782.000 6785.000 6788.000 6796.000 6818.000 6820.000 6823.000 6828.000
6832.000 6840.000 6843.000 6846.000 6849.000 6857.000 6880.000 6883.000 6885.000 6888.000
6893.000 6897.000 6907.000 6913.000 6916.000 6926.000 6937.000 6940.000 6942.000 6945.000
6950.000 6954.000 6964.000 6971.000 6974.000 6984.000 7006.000 7011.000 7018.000 7019.000
7026.000 7036.000 7070.000 7078.000 7081.000 7097.000 7108.000 7114.000 7117.000 7127.000
7142.000 7162.000 7175.000 7180.000 7201.000 7232.000 7280.000 7295.000 7298.000 7301.000
7307.000 7310.000 7321.000 7339.000 7341.000 7344.000 7347.000 7354.000 7381.000 7387.000
7390.000 7396.000 7402.000 7408.000 7410.000 7412.000 7414.000 7442.000 7452.000 7453.000
7454.000 7460.000 7468.000 7475.000 7514.000 7526.000 7537.000 7547.000 7589.000 7598.000
7602.000 7609.000 7617.000 7618.000 7619.000 7620.000 7637.000 7639.000 7661.000 7683.000
7685.000 7701.000 7716.000 7726.000 7746.000 7784.000 7799.000 7816.000 7828.000 7829.000
7833.000 7891.000 7892.000 7952.000 7954.000 7983.000 8008.000 8010.000 8014.000 8016.000
8017.000 8032.000 8036.000 8054.000 8069.000

NOEXTUNIV STRING REFERENCED AT : 639.000 683.000 4501.000 4526.000 4541.000 4546.000 4574.000 5099.000
5142.000 5330.000 5435.000

NON.EXT.IO.REG.LOAD LABEL REFERENCED AT : 1902.000 1927.000

NON.INTERRUPTABLE.EXIT LABEL REFERENCED AT : 2194.000 7358.000 7949.000

NON.PRESENT.MEMORY LABEL REFERENCED AT : 3108.000 3110.000 3115.000 3118.000

NOP LABEL REFERENCED AT : 4524.000 8377.000

NOR LABEL REFERENCED AT : 4583.000 8118.000 8196.000

NOR2 LABEL REFERENCED AT : 4598.000 4590.000

NOR2 STRING REFERENCED AT : 6762.000 6767.000 6785.000 6823.000 6828.000 6846.000 6888.000 6893.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

```
                                    6913.000 6945.000 6950.000 6971.000 7008.000 7070.000 7114.000 7162.000

NOR3                        LABEL   REFERENCED AT :   4585.000 4586.000 4588.000 4591.000

NORC                        STRING  REFERENCED AT :   4588.000 4613.000

NORCNT                      STRING  REFERENCED AT :   6777.000 6794.000 6838.000 6855.000 6962.000 6982.000 7017.000 7125.000

NORD                        LABEL   REFERENCED AT :   4594.000 8119.000 8197.000

NORD2                       LABEL   REFERENCED AT :   4599.000 4602.000

NORD3                       LABEL   REFERENCED AT :   4596.000 4604.000

NORD4                       LABEL   REFERENCED AT :   4601.000 4607.000

NORD5                       LABEL   REFERENCED AT :   4598.000 4600.000 4606.000 4610.000 4613.000

NORD6                       LABEL   REFERENCED AT :   4604.000 4619.000

NORHF                       LABEL   REFERENCED AT :   1756.000 1757.000 1758.000 1760.000 1764.000

NORM                        STRING  REFERENCED AT :   3675.000 4092.000

NORM.EXIT                   LABEL   REFERENCED AT :   3715.000 4142.000

NORM.OFUF                   LABEL   REFERENCED AT :   3532.000 3554.000 3557.000 3559.000

NORMALIZE                   LABEL   REFERENCED AT :   3677.000

NORMCNT                     STRING  REFERENCED AT :   6777.000 6794.000 6838.000 6855.000 6905.000 6924.000 6962.000 6982.000
                                    7017.000 7076.000 7125.000 7173.000

NOT.WCS.EXIT.SYS.RESET      LABEL   REFERENCED AT :   1238.000 5591.000

NP=                         STRING  REFERENCED AT :   8090.000 8093.000 8094.000 8095.000 8096.000 8097.000 8098.000 8099.000
                                    8100.000 8101.000 8103.000 8107.000 8108.000 8109.000 8110.000 8111.000 8120.000 8121.000
                                    8122.000 8123.000 8124.000 8125.000 8127.000 8128.000 8129.000 8130.000 8131.000 8132.000
                                    8133.000 8134.000 8135.000 8137.000 8138.000 8139.000 8140.000 8141.000 8142.000 8143.000
                                    8144.000 8146.000 8147.000 8148.000 8149.000 8150.000 8151.000 8152.000 8153.000 8154.000
                                    8155.000 8156.000 8157.000 8159.000 8168.000 8171.000 8172.000 8173.000 8174.000 8175.000
                                    8176.000 8177.000 8178.000 8179.000 8181.000 8186.000 8187.000 8188.000 8189.000 8198.000
                                    8199.000 8200.000 8201.000 8202.000 8203.000 8546.000 8547.000 8549.000 8581.000 8583.000
                                    8640.000 8644.000 8648.000 8652.000 8663.000 8667.000 8671.000 8675.000

NPM.CHECK                   LABEL   REFERENCED AT :   1001.000 1060.000 1537.000

NU                          STRING  REFERENCED AT :     94.000  594.000  597.000  821.000  822.000  835.000  844.000  846.000
                                     847.000  871.000  946.000 1024.000 1083.000 1087.000 1101.000 1128.000 1150.000 1182.000
                                    1196.000 1204.000 1207.000 1217.000 1220.000 1222.000 1225.000 1228.000 1268.000 1309.000
                                    1327.000 1424.000 1465.000 1468.000 1470.000 1571.000 1724.000 1911.000 1940.000 2067.000
                                    2092.000 2124.000 2127.000 2142.000 2144.000 2152.000 2153.000 2183.000 2201.000 2206.000
                                    2217.000 2235.000 2257.000 2298.000 2422.000 2516.000 2526.000 2533.000 2548.000 2682.000
                                    2967.000 2971.000 2981.000 3167.000 3229.000 3357.000 3421.000 3443.000 3476.000 3518.000
                                    3522.000 3584.000 3640.000 3680.000 3683.000 3764.000 3811.000 3831.000 3848.000 3850.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U


```
                                3852.000 3856.000 3877.000 3894.000 3895.000 3921.000 3922.000 3923.000 3939.000 3966.000
                                3976.000 3997.000 4008.000 4038.000 4044.000 4061.000 4063.000 4068.000 4124.000 4201.000
                                4202.000 4214.000 4264.000 4598.000 4605.000 4609.000 4637.000 4778.000 4789.000 4818.000
                                4827.000 4835.000 4849.000 4871.000 4880.000 4881.000 4897.000 4907.000 4922.000 4944.000
                                4969.000 4998.000 5157.000 5185.000 5194.000 5211.000 5255.000 5311.000 5450.000 5455.000
                                5487.000 5513.000 5612.000 5949.000 6150.000 6394.000 6405.000 6415.000 6417.000 6418.000
                                6474.000 6498.000 6568.000 6588.000 6626.000 6633.000 6993.000 7049.000 7139.000 7279.000
                                7440.000 7441.000 7487.000 7488.000 7496.000 7502.000 7503.000 7509.000 7512.000 7519.000
                                7525.000 7533.000 7534.000 7559.000 7560.000 7569.000 7574.000 7575.000 7582.000 7587.000
                                7605.000 7629.000 7640.000 7657.000 7669.000 7677.000 7810.000 8052.000 8058.000

OF.ROUTE                LABEL   REFERENCED AT :  3547.000 3556.000

OFFSET                  VARIABLE REFERENCED AT :  278.000 1748.000 1748.000 1901.000 2161.000 2166.000 2192.000 2228.000
                                2237.000 2537.000 2608.000 2620.000 2639.000 2639.000 2664.000 5247.000 5247.000 5251.000
                                5252.000 7271.000 7271.000 7321.000 7333.000 7333.000 7340.000 7348.000 7439.000 7439.000
                                7526.000 7598.000 7609.000 7617.000 7637.000 7656.000 7656.000 7658.000 7663.000 7674.000
                                7952.000

OFUF.TEST               LABEL   REFERENCED AT :  3528.000 3716.000

OP.NPM.FLG              VARIABLE REFERENCED AT :  538.000 3108.000 3110.000

OP.PE.FLG               VARIABLE REFERENCED AT :  536.000 3095.000 7820.000

OPERAND.NONPRESENT      LABEL   REFERENCED AT :  740.000  768.000 3104.000

OPERAND.TIMEOUT         LABEL   REFERENCED AT :  3057.000 3105.000 3109.000

OPMIUER                 STRING  REFERENCED AT :  714.000 3062.000 3521.000 7955.000

OPNORESP                STRING  REFERENCED AT :   51.000  210.000  739.000  765.000  944.000  968.000  997.000 1616.000
                                1910.000 1939.000 2050.000 2681.000 3053.000 3227.000 3419.000 3521.000 4636.000 4785.000
                                4817.000 4826.000 4834.000 4848.000 4879.000 4896.000 4918.000 4943.000 5193.000 5301.000
                                5309.000 5466.000 5504.000 5510.000 5531.000 5905.000 5912.000 6340.000 6497.000 6567.000
                                6587.000 6694.000 6745.000 7828.000 7955.000 8059.000

OPR1.GT.OPR2.GT6        LABEL   REFERENCED AT :  3601.000 3634.000

OPR1.LT.OPR2.GT6        LABEL   REFERENCED AT :  3600.000 3636.000

OPRND.NONPRESENT        LABEL   REFERENCED AT :  3054.000 3106.000

OPRND.PE                LABEL   REFERENCED AT :  715.000  771.000 3060.000 3094.000

OPRNDPE                 STRING  REFERENCED AT :   51.000  210.000  769.000  944.000  968.000  997.000 1582.000 1616.000
                                1910.000 1939.000 2050.000 2681.000 3059.000 3227.000 3419.000 4636.000 4785.000 4817.000
                                4826.000 4834.000 4848.000 4879.000 4896.000 4918.000 4943.000 5193.000 5301.000 5309.000
                                5466.000 5531.000 5912.000 6497.000 6567.000 6587.000 6694.000 6745.000 7828.000 7955.000
                                8059.000

OPTIMEOUT               STRING  REFERENCED AT :   51.000  210.000  739.000  765.000  944.000  968.000  997.000 1616.000
                                1910.000 1939.000 2050.000 2681.000 3056.000 3105.000 3227.000 3419.000 3521.000 4636.000
                                4785.000 4817.000 4826.000 4834.000 4848.000 4879.000 4896.000 4918.000 4943.000 5193.000
                                5301.000 5309.000 5466.000 5531.000 5903.000 5912.000 6497.000 6567.000 6587.000 6745.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

		7826.000 7955.000 8059.000
OTHERBANK	STRING	REFERENCED AT : 816.000 864.000 865.000 870.000 905.000 908.000 909.000 913.000 932.000 936.000 957.000 960.000 961.000 1125.000 1196.000 1204.000 1216.000 1235.000 1255.000 1270.000 1290.000 1291.000 1346.000 1353.000 1368.000 1370.000 1532.000 1534.000 1594.000 1606.000 1607.000 1610.000 1627.000 1675.000 1677.000 1681.000 1682.000 1684.000 1818.000 1821.000 1823.000 1855.000 1904.000 2027.000 2039.000 2496.000 2497.000 2531.000 2551.000 2552.000 2560.000 2688.000 3153.000 3155.000 3162.000 3172.000 3173.000 3174.000 3214.000 3215.000 3221.000 3305.000 3316.000 3320.000 3323.000 3351.000 3388.000 3422.000 3423.000 3424.000 3427.000 3504.000 3505.000 3519.000 3533.000 3537.000 3540.000 3583.000 3649.000 3764.000 3768.000 3780.000 3795.000 3797.000 3803.000 3805.000 3822.000 3827.000 3828.000 3849.000 3898.000 3911.000 3912.000 3913.000 3931.000 3932.000 3942.000 3943.000 4024.000 4141.000 4183.000 4184.000 4185.000 4301.000 4305.000 4308.000 5242.000 5308.000 5323.000 5349.000 5351.000 5366.000 5727.000 5934.000 6029.000 6091.000 6092.000 6096.000 6098.000 6099.000 6101.000 6157.000 6158.000 6424.000 6465.000 6480.000 6500.000 6506.000 6548.000 6565.000 6589.000 6590.000 6609.000 7205.000 7246.000 7250.000 7750.000 8041.000 8042.000
OTHERS.TEST	LABEL	REFERENCED AT : 697.000 1071.000
OVERFLOW	LABEL	REFERENCED AT : 3558.000 3561.000 3569.000
OVFMOPEZ	STRING	REFERENCED AT : 7053.000 7145.000
PANEL.ATTN	LABEL	REFERENCED AT : 1033.000
PANEL.EVALUATE.PSD2	LABEL	REFERENCED AT : 1656.000 5342.000 6003.000
PANEL.HALT	LABEL	REFERENCED AT : 1012.000 1016.000 1017.000 1025.000 1041.000 3327.000 6339.000 6341.000
PANEL.SYS.RES	LABEL	REFERENCED AT : 1011.000 1019.000
PC	STRING	REFERENCED AT : 896.000 953.000 1102.000 1114.000 1118.000 1138.000 1231.000 1675.000 1765.000 1838.000 2552.000 3266.000 3271.000 3304.000 5069.000 5094.000 5115.000 5138.000 5140.000 5196.000 5308.000 5731.000 6096.000 6701.000 7750.000
PCMASK	VARIABLE	REFERENCED AT : 153.000 864.000 936.000 953.000 1168.000 1607.000 1675.000 1838.000 1904.000 2039.000 2552.000 2688.000 3305.000 5112.000 5305.000 5727.000 5934.000 6029.000 6092.000 6096.000 7750.000
PCTOMAR	STRING	REFERENCED AT : 673.000 858.000 860.000 931.000 937.000 1105.000 1120.000 1135.000 1233.000 1523.000 1754.000 3068.000 3137.000 3176.000 3202.000 3232.000 3304.000 3353.000 3416.000 4550.000 5116.000 6009.000 6090.000 6097.000 6314.000 6316.000 6319.000 7464.000 7766.000 7855.000 7909.000
PF.QUEUE	LABEL	REFERENCED AT : 1129.000 3169.000
PHY	VARIABLE	REFERENCED AT : 280.000
PLUSONE	STRING	REFERENCED AT : 7080.000 7179.000
PNL.WRK	VARIABLE	REFERENCED AT : 291.000 5870.000 5871.000 6165.000 6171.000 6177.000 6185.000
PNLDATA	REGISTER	REFERENCED AT : 5748.000 5749.000 5954.000 5955.000 6139.000 6148.000 6266.000

MICROCODE CROSS-REFFRENCE LIST

SEL 32/75 CPU

POST.NORM	LABEL	REFERENCED AT :	3651.000 3669.000 3879.000 3881.000 3907.000
POWER.FAIL	LABEL	REFERENCED AT :	674.000 1099.000
POWER.OFF	LABEL	REFERENCED AT :	1106.000
POWER.ON	LABEL	REFERENCED AT :	1105.000 1115.000 1120.000
POWER.READ.OK	LABEL	REFERENCED AT :	1117.000 6696.000
POWER.UP.ERROR	LABEL	REFERENCED AT :	6695.000 7803.000
POWER.UP.EXIT	LABEL	REFERENCED AT :	7809.000 7814.000
POWER.UP.S.P.CLEAR	LABEL	REFERENCED AT :	1121.000 6692.000 7804.000
PP.CSW.ADDR	LABEL	REFERENCED AT :	1021.000 1328.000 1711.000 7807.000
PP.CSW.ADDR1	LABEL	REFERENCED AT :	1022.000 1329.000 1700.000 7808.000
PP.END	LABEL	REFERENCED AT :	1514.000 1622.000 1626.000 1657.000 1665.000 1691.000
PP.NPRESENT	LABEL	REFERENCED AT :	1519.000 1623.000 1646.000
PP.PRESENT	LABEL	REFERENCED AT :	1645.000 1647.000
PP.WT.PSD2	LABEL	REFERENCED AT :	1654.000
PPATTN	STRING	REFERENCED AT :	1016.000
PPATTN.HALT1	LABEL	REFERENCED AT :	1038.000 1509.000
PPATTN.HALT2	LABEL	REFERENCED AT :	1037.000 1511.000
PPATTN.RD	LABEL	REFERENCED AT :	1524.000 1548.000
PPATTN.RD.FILE	LABEL	REFERENCED AT :	1554.000 1592.000
PPATTN.RD.M	LABEL	REFERENCED AT :	1580.000 1583.000 1585.000 1594.000 1615.000
PPATTN.RD.M1	LABEL	REFERENCED AT :	1557.000 1619.000 1727.000
PPATTN.RD.PSW	LABEL	REFERENCED AT :	1552.000 1605.000 6319.000
PPATTN.WT	LABEL	REFERENCED AT :	1523.000 1641.000
PPATTN.WT.CONV	LABEL	REFERENCED AT :	1652.000 1722.000
PPATTN.WT.PSW	LABEL	REFERENCED AT :	1651.000 1674.000 6311.000
PPATTN.WT.REG	LABEL	REFERENCED AT :	1653.000 1663.000
PPINST.STEP	LABEL	REFERENCED AT :	1531.000
PPREAD.PSD2	LABEL	REFERENCED AT :	1555.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

PRIV	STRING	REFERENCED AT : 800.000 1677.000 1680.000 2560.000 2563.000 3224.000 5308.000 5313.000
PRIV.ERROR	LABEL	REFERENCED AT : 5411.000 5413.000
PRIV.ERRORS	LABEL	REFERENCED AT : 5389.000 5393.000 5402.000 6806.000
PRIV.EXECUTE.ERROR	LABEL	REFERENCED AT : 4411.000
PRIV.TYPE	LABEL	REFERENCED AT : 1654.000 3128.000
PRIV.VIOL	LABEL	REFERENCED AT : 2523.000 4413.000 4458.000 5024.000 5276.000 5394.000 5414.000 5542.000
PRIVBIT	STRING	REFERENCED AT : 2522.000 4410.000 4412.000 4457.000 5023.000 5389.000 5402.000 5411.000 5541.000 6805.000
PRIVILEGE.VIOLATION	LABEL	REFERENCED AT : 742.000 776.000 1075.000 3123.000
PROG.VIOLATION	LABEL	REFERENCED AT : 2273.000 2294.000 2410.000 2411.000
PROT.MAP.CLEAR	LABEL	REFERENCED AT : 1183.000 1186.000
PROTV	STRING	REFERENCED AT : 741.000 774.000 1730.000 4907.000 7827.000
PSD.ADDR	VARIABLE	REFERENCED AT : 231.000 1835.000 2530.000
PSW1	VARIABLE	REFERENCED AT : 214.000 2535.000 2560.000 5307.000 5323.000 7715.000 7747.000
PUSHJ	STRING	REFERENCED AT : 5575.000
PWR.F.UP.FLG	VARIABLE	REFERENCED AT : 552.000 7821.000
PWR.OFF.OUT	LABEL	REFERENCED AT : 1114.000 1122.000 2101.000 7819.000
PWRFAIL	STRING	REFERENCED AT : 673.000
Q3E	LABEL	REFERENCED AT : 5012.000 8157.000
QUEUE.CALM	LABEL	REFERENCED AT : 5247.000 5251.000 5252.000 5256.000
QUEUE.EXIT	LABEL	REFERENCED AT : 5257.000 5258.000 5260.000
QUEUE.TEST	LABEL	REFERENCED AT : 1795.000 1858.000 3103.000 3130.000 3147.000 3151.000
QUEUE.TEST.BRANCH	LABEL	REFERENCED AT : 3122.000 3129.000
R	STRING	REFERENCED AT : 705.000 2493.000 3479.000 3516.000 3540.000 3572.000 3574.000 3583.000 3648.000 3762.000 3765.000 3780.000 3781.000 3782.000 3849.000 3898.000 3912.000 3913.000 4184.000 4185.000 4410.000 4476.000 4480.000 4482.000 4487.000 4488.000 4488.000 4492.000 4493.000 4497.000 4499.000 4507.000 4508.000 4514.000 4514.000 4515.000 4516.000 4518.000 4519.000 4520.000 4531.000 4550.000 4553.000 4553.000 4555.000 4560.000 4568.000 4572.000 4577.000 4578.000 4583.000 4591.000 4594.000 4615.000 4649.000 4657.000 4657.000 4661.000 4670.000 4674.000 4687.000 4704.000 4708.000 4708.000 4712.000 4713.000 4715.000 4715.000 4727.000 4731.000 4733.000 4746.000 4752.000 4760.000 4767.000 4895.000 4941.000 4950.000 4952.000 5077.000 5125.000 5126.000 5126.000 5127.000 5127.000 5128.000 5128.000 5129.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

5129.000 5133.000 5135.000 5403.000 5412.000 5466.000 5500.000 5521.000 6666.000 6672.000
6674.000 6684.000 6684.000 6752.000 6771.000 6774.000 6779.000 6788.000 6791.000 6796.000
6813.000 6832.000 6835.000 6840.000 6849.000 6852.000 6857.000 6876.000 6897.000 6901.000
6907.000 6916.000 6920.000 6926.000 6933.000 6954.000 6958.000 6964.000 6974.000 6978.000
6984.000 6989.000 7011.000 7014.000 7019.000 7045.000 7059.000 7074.000 7078.000 7081.000
7084.000 7090.000 7117.000 7121.000 7127.000 7134.000 7151.000 7175.000 7180.000 7184.000
7205.000 7891.000 7892.000 7985.000 8008.000 8010.000 8014.000 8016.000 8017.000 8032.000
8036.000

R(STRING REFERENCED AT : 672.000 676.000 677.000 678.000 698.000 705.000 785.000 800.000
801.000 812.000 816.000 818.000 824.000 859.000 864.000 866.000 869.000 870.000
897.000 906.000 913.000 929.000 936.000 937.000 947.000 953.000 957.000 960.000
961.000 970.000 997.000 997.000 1004.000 1015.000 1059.000 1064.000 1072.000 1074.000
1089.000 1114.000 1154.000 1184.000 1197.000 1232.000 1233.000 1257.000 1279.000 1284.000
1295.000 1300.000 1301.000 1302.000 1305.000 1305.000 1310.000 1332.000 1333.000 1337.000
1346.000 1355.000 1358.000 1368.000 1370.000 1374.000 1375.000 1380.000 1382.000 1384.000
1387.000 1389.000 1390.000 1391.000 1396.000 1396.000 1399.000 1405.000 1406.000 1425.000
1426.000 1427.000 1431.000 1436.000 1456.000 1458.000 1459.000 1459.000 1461.000 1466.000
1517.000 1520.000 1532.000 1533.000 1549.000 1549.000 1575.000 1593.000 1594.000 1606.000
1607.000 1610.000 1648.000 1648.000 1664.000 1665.000 1675.000 1681.000 1746.000 1775.000
1811.000 1814.000 1818.000 1819.000 1835.000 1838.000 1840.000 1890.000 1893.000 1898.000
1904.000 1906.000 1908.000 1931.000 1936.000 1937.000 2027.000 2040.000 2041.000 2042.000
2043.000 2055.000 2065.000 2066.000 2070.000 2125.000 2161.000 2166.000 2179.000 2181.000
2184.000 2192.000 2202.000 2204.000 2249.000 2255.000 2256.000 2261.000 2266.000 2272.000
2275.000 2289.000 2299.000 2308.000 2310.000 2317.000 2331.000 2341.000 2364.000 2369.000
2370.000 2371.000 2372.000 2377.000 2379.000 2381.000 2394.000 2397.000 2402.000 2423.000
2426.000 2446.000 2461.000 2465.000 2466.000 2467.000 2468.000 2470.000 2471.000 2476.000
2493.000 2495.000 2497.000 2522.000 2527.000 2530.000 2531.000 2532.000 2535.000 2536.000
2540.000 2550.000 2560.000 2566.000 2573.000 2584.000 2593.000 2605.000 2618.000 2662.000
2675.000 2677.000 2678.000 2684.000 2686.000 2690.000 2693.000 2696.000 2710.000 2713.000
2733.000 2740.000 2763.000 2767.000 2772.000 2775.000 2825.000 2860.000 2871.000 2881.000
2886.000 2897.000 2898.000 2914.000 2915.000 2916.000 2926.000 2930.000 2938.000 2939.000
2941.000 2945.000 2957.000 2968.000 3152.000 3162.000 3170.000 3215.000 3220.000 3221.000
3224.000 3226.000 3231.000 3309.000 3320.000 3323.000 3350.000 3351.000 3388.000 3390.000
3411.000 3424.000 3427.000 3450.000 3460.000 3472.000 3474.000 3479.000 3503.000 3516.000
3519.000 3533.000 3537.000 3540.000 3572.000 3574.000 3583.000 3595.000 3599.000 3611.000
3633.000 3635.000 3641.000 3648.000 3651.000 3660.000 3683.000 3690.000 3694.000 3699.000
3702.000 3730.000 3731.000 3736.000 3744.000 3750.000 3757.000 3762.000 3763.000 3765.000
3766.000 3768.000 3780.000 3781.000 3782.000 3783.000 3784.000 3794.000 3796.000 3797.000
3802.000 3803.000 3804.000 3806.000 3808.000 3809.000 3813.000 3816.000 3817.000 3818.000
3622.000 3826.000 3831.000 3832.000 3836.000 3846.000 3847.000 3849.000 3850.000 3854.000
3857.000 3877.000 3889.000 3891.000 3894.000 3898.000 3905.000 3906.000 3910.000 3911.000
3912.000 3913.000 3915.000 3916.000 3917.000 3919.000 3921.000 3922.000 3929.000 3930.000
3931.000 3932.000 3934.000 3936.000 3937.000 3938.000 3942.000 3943.000 3961.000 3962.000
3964.000 3965.000 3966.000 3968.000 3973.000 3981.000 3985.000 3994.000 3997.000 3998.000
3999.000 4002.000 4003.000 4024.000 4024.000 4025.000 4026.000 4027.000 4040.000 4041.000
4044.000 4045.000 4046.000 4055.000 4064.000 4066.000 4068.000 4073.000 4075.000 4077.000
4078.000 4084.000 4085.000 4091.000 4140.000 4147.000 4183.000 4184.000 4185.000 4191.000
4192.000 4193.000 4195.000 4197.000 4201.000 4205.000 4206.000 4207.000 4208.000 4209.000
4210.000 4214.000 4215.000 4219.000 4222.000 4226.000 4229.000 4229.000 4231.000 4231.000
4233.000 4234.000 4235.000 4244.000 4246.000 4248.000 4251.000 4255.000 4258.000 4260.000
4264.000 4272.000 4273.000 4288.000 4289.000 4292.000 4293.000 4294.000 4295.000 4296.000
4297.000 4298.000 4301.000 4305.000 4306.000 4368.000 4369.000 4370.000 4371.000 4372.000
4373.000 4374.000 4375.000 4377.000 4380.000 4409.000 4410.000 4464.000 4476.000 4480.000
4482.000 4487.000 4488.000 4488.000 4492.000 4493.000 4497.000 4498.000 4499.000 4499.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

4500.000 4507.000 4508.000 4508.000 4511.000 4514.000 4514.000 4515.000 4516.000 4517.000
4518.000 4519.000 4519.000 4520.000 4531.000 4531.000 4537.000 4537.000 4539.000 4544.000
4545.000 4545.000 4550.000 4553.000 4553.000 4555.000 4560.000 4568.000 4569.000 4571.000
4572.000 4577.000 4578.000 4583.000 4585.000 4586.000 4590.000 4590.000 4591.000 4594.000
4595.000 4602.000 4614.000 4615.000 4619.000 4630.000 4634.000 4649.000 4655.000 4657.000
4657.000 4661.000 4670.000 4674.000 4687.000 4695.000 4704.000 4708.000 4708.000 4712.000
4713.000 4715.000 4715.000 4727.000 4731.000 4733.000 4741.000 4746.000 4752.000 4760.000
4767.000 4768.000 4781.000 4782.000 4796.000 4816.000 4825.000 4833.000 4846.000 4861.000
4869.000 4870.000 4873.000 4877.000 4895.000 4911.000 4912.000 4941.000 4950.000 4952.000
4962.000 4966.000 4991.000 4995.000 5013.000 5025.000 5025.000 5027.000 5040.000 5061.000
5077.000 5089.000 5111.000 5114.000 5116.000 5124.000 5125.000 5126.000 5126.000 5127.000
5127.000 5128.000 5128.000 5129.000 5129.000 5132.000 5133.000 5135.000 5158.000 5162.000
5185.000 5186.000 5191.000 5205.000 5222.000 5223.000 5224.000 5245.000 5246.000 5251.000
5252.000 5299.000 5307.000 5323.000 5349.000 5366.000 5403.000 5411.000 5412.000 5456.000
5458.000 5466.000 5499.000 5500.000 5521.000 5522.000 5570.000 5727.000 5870.000 5871.000
5934.000 6029.000 6092.000 6096.000 6097.000 6101.000 6155.000 6157.000 6165.000 6171.000
6177.000 6185.000 6382.000 6424.000 6442.000 6465.000 6476.000 6480.000 6483.000 6484.000
6499.000 6505.000 6516.000 6540.000 6548.000 6585.000 6586.000 6594.000 6609.000 6642.000
6654.000 6660.000 6666.000 6672.000 6674.000 6677.000 6684.000 6684.000 6701.000 6719.000
6728.000 6752.000 6771.000 6774.000 6779.000 6788.000 6791.000 6796.000 6813.000 6832.000
6835.000 6840.000 6849.000 6852.000 6857.000 6875.000 6876.000 6897.000 6898.000 6901.000
6902.000 6907.000 6908.000 6916.000 6917.000 6920.000 6921.000 6926.000 6927.000 6932.000
6933.000 6954.000 6955.000 6958.000 6959.000 6964.000 6965.000 6974.000 6975.000 6978.000
6979.000 6984.000 6985.000 6989.000 7011.000 7014.000 7019.000 7045.000 7059.000 7074.000
7078.000 7081.000 7084.000 7089.000 7090.000 7117.000 7118.000 7121.000 7122.000 7127.000
7128.000 7133.000 7134.000 7151.000 7169.000 7175.000 7176.000 7180.000 7181.000 7184.000
7185.000 7205.000 7239.000 7244.000 7244.000 7245.000 7250.000 7270.000 7339.000 7340.000
7347.000 7349.000 7351.000 7352.000 7414.000 7451.000 7460.000 7462.000 7475.000 7477.000
7483.000 7484.000 7491.000 7526.000 7527.000 7536.000 7538.000 7547.000 7549.000 7555.000
7556.000 7563.000 7596.000 7598.000 7601.000 7609.000 7626.000 7627.000 7650.000 7653.000
7658.000 7701.000 7715.000 7747.000 7881.000 7891.000 7892.000 7980.000 7984.000 7985.000
8008.000 8010.000 8014.000 8016.000 8017.000 8029.000 8032.000 8034.000 8036.000 8046.000
8049.000 8061.000

R.NOR.ADD.DOUBLE.1 LABEL REFERENCED AT : 6904.000

R.NOR.ADD.DOUBLE.2 LABEL REFERENCED AT : 6923.000

R.NOR.ADD.SING.1 LABEL REFERENCED AT : 6776.000

R.NOR.ADD.SING.2 LABEL REFERENCED AT : 6793.000

R.NOR.DIVIDE.DOUBLE.1 LABEL REFERENCED AT : 7172.000

R.NOR.MULTIPLY.DOUBLE.1 LABEL REFERENCED AT : 7124.000

R.NOR.MULTIPLY.SING.1 LABEL REFERENCED AT : 7016.000

R.NOR.SUB.DOUBLE.1 LABEL REFERENCED AT : 6961.000

R.NOR.SUB.DOUBLE.2 LABEL REFERENCED AT : 6981.000

R.NOR.SUB.SING.1 LABEL REFERENCED AT : 6837.000

R.NOR.SUB.SING.2 LABEL REFERENCED AT : 6854.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

R.RND.ADD.DOUBLE.1	LABEL	REFERENCED AT :	6896.000
R.RND.ADD.DOUBLE.2	LABEL	REFERENCED AT :	6915.000
R.RND.ADD.SING.1	LABEL	REFERENCED AT :	6770.000
R.RND.ADD.SING.2	LABEL	REFERENCED AT :	6787.000
R.RND.DIVIDE.DOUBLE.1	LABEL	REFERENCED AT :	7164.000
R.RND.MULTIPLY.DOUBLE.1	LABEL	REFERENCED AT :	7116.000
R.RND.MULTIPLY.SING.1	LABEL	REFERENCED AT :	7010.000
R.RND.SUB.DOUBLE.1	LABEL	REFERENCED AT :	6953.000
R.RND.SUB.DOUBLE.2	LABEL	REFERENCED AT :	6973.000
R.RND.SUB.SING.1	LABEL	REFERENCED AT :	6831.000
R.RND.SUB.SING.2	LABEL	REFERENCED AT :	6848.000
R.ZERO.ADD.DOUBLE.1	LABEL	REFERENCED AT :	6900.000
R.ZERO.ADD.DOUBLE.2	LABEL	REFERENCED AT :	6919.000
R.ZERO.ADD.SING.1	LABEL	REFERENCED AT :	6773.000
R.ZERO.ADD.SING.2	LABEL	REFERENCED AT :	6790.000
R.ZERO.DIVIDE.DOUBLE.1	LABEL	REFERENCED AT :	7168.000
R.ZERO.MULTIPLY.DOUBLE.1	LABEL	REFERENCED AT :	7120.000
R.ZERO.MULTIPLY.SING.1	LABEL	REFERENCED AT :	7013.000
R.ZERO.SUB.DOUBLE.1	LABEL	REFERENCED AT :	6957.000
R.ZERO.SUB.DOUBLE.2	LABEL	REFERENCED AT :	6977.000
R.ZERO.SUB.SING.1	LABEL	REFERENCED AT :	6834.000
R.ZERO.SUB.SING.2	LABEL	REFERENCED AT :	6851.000
R00000000	LABEL	REFERENCED AT :	3678.000 3891.000 3898.000
R0000000X	LABEL	REFERENCED AT :	3681.000
R000000XX	LABEL	REFERENCED AT :	3688.000
R00000XXX	LABEL	REFERENCED AT :	3692.000
R0000XXXX	LABEL	REFERENCED AT :	3697.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

R000XXXXX LABEL REFERENCED AT : 3700.000

R00XXXXXX LABEL REFERENCED AT : 3680.000 3683.000 3705.000 3706.000 3708.000 3712.000

R0XXXXXXX LABEL REFERENCED AT : 3685.000 3703.000

RA STRING REFERENCED AT : 6753.000 6777.000 6794.000 6814.000 6838.000 6855.000 6877.000 6905.000
6924.000 6934.000 6962.000 6982.000 6991.000 6999.000 7001.000 7004.000 7005.000 7017.000
7047.000 7054.000 7057.000 7065.000 7067.000 7076.000 7083.000 7091.000 7102.000 7104.000
7105.000 7110.000 7125.000 7137.000 7146.000 7149.000 7157.000 7159.000 7173.000 7183.000

RA REGISTER REFERENCED AT : 6757.000 6759.000 6762.000 6767.000 6774.000 6777.000 6779.000 6782.000
6785.000 6791.000 6794.000 6796.000 6818.000 6820.000 6823.000 6828.000 6835.000 6838.000
6840.000 6843.000 6846.000 6852.000 6855.000 6857.000 6883.000 6885.000 6888.000 6893.000
6897.000 6898.000 6901.000 6902.000 6905.000 6907.000 6908.000 6913.000 6916.000 6917.000
6920.000 6921.000 6924.000 6926.000 6927.000 6940.000 6942.000 6945.000 6950.000 6954.000
6955.000 6958.000 6959.000 6962.000 6964.000 6965.000 6971.000 6974.000 6975.000 6978.000
6979.000 6982.000 6984.000 6985.000 6991.000 6996.000 6998.000 7001.000 7002.000 7003.000
7005.000 7008.000 7014.000 7017.000 7018.000 7019.000 7047.000 7052.000 7054.000 7057.000
7065.000 7067.000 7070.000 7076.000 7078.000 7084.000 7101.000 7104.000 7105.000 7108.000
7110.000 7114.000 7117.000 7118.000 7121.000 7122.000 7125.000 7127.000 7128.000 7137.000
7144.000 7146.000 7149.000 7157.000 7159.000 7162.000 7173.000 7175.000 7176.000 7180.000
7181.000 7184.000

RAE STRING REFERENCED AT : 6753.000 6759.000 6767.000 6777.000 6778.000 6782.000 6794.000 6795.000
6814.000 6820.000 6828.000 6838.000 6839.000 6843.000 6855.000 6856.000 6877.000 6885.000
6893.000 6905.000 6906.000 6911.000 6924.000 6925.000 6934.000 6942.000 6950.000 6962.000
6963.000 6968.000 6982.000 6983.000 6995.000 6996.000 6998.000 7017.000 7018.000 7051.000
7052.000 7053.000 7080.000 7098.000 7099.000 7101.000 7125.000 7126.000 7143.000 7144.000
7145.000 7179.000

RAE REGISTER REFERENCED AT : 6767.000 6771.000 6777.000 6778.000 6779.000 6782.000 6788.000 6794.000
6795.000 6796.000 6828.000 6832.000 6838.000 6839.000 6840.000 6843.000 6849.000 6855.000
6856.000 6857.000 6893.000 6897.000 6905.000 6906.000 6907.000 6911.000 6916.000 6924.000
6925.000 6926.000 6950.000 6954.000 6962.000 6963.000 6964.000 6968.000 6974.000 6982.000
6983.000 6984.000 6996.000 6998.000 7011.000 7017.000 7018.000 7019.000 7052.000 7053.000
7063.000 7078.000 7080.000 7081.000 7099.000 7101.000 7117.000 7125.000 7126.000 7127.000
7144.000 7145.000 7155.000 7175.000 7179.000 7180.000

RAM.LOAD LABEL REFERENCED AT : 926.000 1808.000 2067.000 2085.000 2183.000 2257.000

RAM.LOAD.EXIT LABEL REFERENCED AT : 2089.000 2091.000

RB STRING REFERENCED AT : 6756.000 6817.000 6882.000 6939.000 6990.000 6992.000 6995.000 6996.000
7002.000 7046.000 7051.000 7052.000 7055.000 7061.000 7063.000 7066.000 7096.000 7098.000
7135.000 7136.000 7143.000 7144.000 7147.000 7153.000 7155.000 7158.000

RB REGISTER REFERENCED AT : 6757.000 6759.000 6818.000 6820.000 6883.000 6885.000 6940.000 6942.000
6991.000 6996.000 7003.000 7047.000 7052.000 7055.000 7061.000 7063.000 7067.000 7108.000
7136.000 7137.000 7144.000 7147.000 7153.000 7155.000 7159.000 7185.000

RBE STRING REFERENCED AT : 6990.000 6991.000 7046.000 7047.000 7091.000 7092.000 7135.000 7137.000

RBE REGISTER REFERENCED AT : 6905.000 6924.000 6991.000 6998.000 7047.000 7053.000 7063.000 7065.000
7092.000 7101.000 7137.000 7145.000 7155.000 7157.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

```
RD                          STRING   REFERENCED AT :  4870.000 4877.000 6752.000 6753.000 6756.000 6813.000 6814.000 6817.000
                                     6875.000 6876.000 6877.000 6882.000 6932.000 6933.000 6934.000 6939.000 6989.000 6990.000
                                     6995.000 6998.000 6999.000 7001.000 7004.000 7045.000 7046.000 7051.000 7059.000 7065.000
                                     7066.000 7083.000 7089.000 7090.000 7091.000 7098.000 7101.000 7102.000 7104.000 7133.000
                                     7134.000 7135.000 7143.000 7151.000 7157.000 7158.000 7183.000

RD.EFF.ERR                  LABEL    REFERENCED AT :  1576.000 1581.000

RDEV                        VARIABLE REFERENCED AT :   260.000  812.000  814.000  824.000  905.000  929.000 1269.000 1279.000
                                     1284.000 1295.000 1337.000 1515.000 1643.000 1714.000 1775.000 1810.000 1893.000 2086.000
                                     2125.000 2179.000 2202.000 2204.000 2261.000 2262.000 2272.000 2274.000 2290.000 2423.000
                                     2426.000 2460.000 2540.000 2675.000 2677.000 2686.000 2699.000 2727.000 2732.000 2741.000
                                     2742.000 2745.000 2746.000 2881.000 2914.000 2926.000 2938.000 3230.000 3320.000 7232.000
                                     7244.000 7244.000 7423.000 7491.000 7563.000 7650.000 7653.000 7693.000 7693.000

RDEXP                       STRING   REFERENCED AT :  6753.000 6814.000 6877.000 6934.000 6990.000 7046.000 7091.000 7135.000

RDLOCK                      STRING   REFERENCED AT :  4963.000 4967.000

RDLSW                       STRING   REFERENCED AT :  6745.000 6881.000 6882.000 6938.000 6939.000 7096.000 7097.000 7141.000
                                     7142.000

RDMAP                       STRING   REFERENCED AT :  1726.000 5466.000 5978.000 6666.000 6671.000

RDMS                        STRING   REFERENCED AT :  7003.000

RDSTS                       LABEL    REFERENCED AT :  7976.000 8384.000

RDY.HERE                    LABEL    REFERENCED AT :  2133.000 2135.000 2139.000

RDY.TIM.FLG                 VARIABLE REFERENCED AT :   533.000 5030.000 7509.000 7582.000 7666.000

READ                        STRING   REFERENCED AT :   943.000 1510.000 1617.000 1838.000 1900.000 1906.000 1933.000 1954.000
                                     2015.000 2042.000 2665.000 2690.000 3101.000 3120.000 3212.000 3226.000 3411.000 3418.000
                                     3516.000 3520.000 3534.000 3914.000 4026.000 4192.000 4472.000 4631.000 4635.000 4782.000
                                     4813.000 4822.000 4830.000 4992.000 4996.000 5296.000 5300.000 5307.000 5454.000 5523.000
                                     5526.000 5895.000 6474.000 6482.000 6516.000 6566.000 6586.000 6693.000 6729.000 6752.000
                                     6813.000 6875.000 6879.000 6932.000 6936.000 6989.000 7045.000 7089.000 7093.000 7134.000
                                     7136.000 8045.000 8050.000

READ.EFFECTIVE.ADDR         LABEL    REFERENCED AT :  1553.000 1569.000 6011.000

READ.WCS                    LABEL    REFERENCED AT :  7798.000

READY.EXIT                  LABEL    REFERENCED AT :  7661.000 7671.000

REFRFSH.RLK.TIMEOUT         LABEL    REFERENCED AT :  5345.000 5369.000

REGRFG                      LABEL    REFERENCED AT :  4511.000 8093.000 8094.000 8095.000 8096.000 8097.000 8103.000 8108.000
                                     8109.000 8171.000 8172.000 8173.000 8174.000 8175.000 8181.000 8186.000 8187.000

RENABLE.EXIT                LABEL    REFERENCED AT :   693.000 1080.000

REPEAT                      STRING   REFERENCED AT :  3772.000 3876.000 4042.000 4067.000 5489.000
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

REPLACE	LABEL	REFERENCED AT : 3596.000 3638.000
REQ.INTR	LABEL	REFERENCED AT : 1749.000 2595.000 5261.000
REQUEST.IT	LABEL	REFERENCED AT : 1732.000 1735.000 1744.000
RESET(	STRING	REFERENCED AT : 410.000 639.000 683.000 692.000 800.000 813.000 865.000 951.000 966.000 983.000 985.000 1003.000 1031.000 1058.000 1063.000 1083.000 1092.000 1097.000 1100.000 1104.000 1127.000 1132.000 1133.000 1150.000 1181.000 1274.000 1464.000 1496.000 1498.000 1534.000 1570.000 1629.000 1655.000 1677.000 1687.000 1708.000 1726.000 1778.000 1781.000 1790.000 1823.000 1836.000 1856.000 1954.000 2167.000 2241.000 2279.000 2413.000 2498.000 2551.000 2560.000 2561.000 2640.000 2716.000 3065.000 3066.000 3132.000 3140.000 3167.000 3179.000 3205.000 3210.000 3224.000 3274.000 3314.000 3384.000 3498.000 3649.000 3731.000 3736.000 3743.000 3780.000 3834.000 4399.000 4420.000 4442.000 4475.000 4524.000 5098.000 5141.000 5268.000 5271.000 5296.000 5308.000 5315.000 5317.000 5329.000 5330.000 5347.000 5352.000 5372.000 5391.000 5417.000 5440.000 5472.000 5502.000 5507.000 5512.000 5533.000 5536.000 5540.000 5577.000 5578.000 5614.000 5823.000 5859.000 5980.000 6002.000 6215.000 6245.000 6301.000 6302.000 6385.000 6426.000 6432.000 6549.000 6608.000 6645.000 6672.000 6680.000 6719.000 7037.000 7169.000 7334.000 7355.000 7452.000 7620.000 7624.000 7685.000 7686.000 7727.000 7750.000 7785.000 7800.000 7838.000 7843.000 7849.000 7900.000 7930.000 7952.000 7954.000 8008.000 8025.000 8026.000 8038.000 8043.000
RESET.HIREG.RTN	LABEL	REFERENCED AT : 1703.000 1707.000 2628.000 2633.000
RESET.INTERRUPT.ACTIVE	LABEL	REFERENCED AT : 7618.000 7623.000
RESET.MAP	LABEL	REFERENCED AT : 1234.000
RESETFF	STRING	REFERENCED AT : 1151.000 1165.000 1268.000 1327.000
RESETIO	STRING	REFERENCED AT : 1103.000
RESPONSE.BUT.NOT.READY	LABEL	REFERENCED AT : 7665.000
RESPONSE.ON.BUS	LABEL	REFERENCED AT : 7411.000 7654.000
RESTORE.CPU.MODE	LABEL	REFERENCED AT : 3067.000 3381.000 7765.000
RESTORE.SCRATCH	LABEL	REFERENCED AT : 1256.000 1265.000
RETRY.AICT	LABEL	REFERENCED AT : 2203.000 2209.000
RETRY.ARSTX	LABEL	REFERENCED AT : 1275.000 1516.000 1644.000 2121.000 2123.000 2130.000
RETRY.READY	LABEL	REFERENCED AT : 7504.000 7508.000
RETRY.READY1	LABEL	REFERENCED AT : 7576.000 7581.000
RH	STRING	REFERENCED AT : 4708.000
RHFLAG	STRING	REFERENCED AT : 1135.000 1255.000 1760.000 3267.000 4469.000 4500.000 4525.000 4540.000 4545.000 4573.000 5078.000 5243.000 5544.000
RI	LABEL	REFERENCED AT : 2592.000 8571.000 8572.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

RI.EXIT	LABEL	REFERENCED AT : 2597.000 2598.000
RI.PENDING	LABEL	REFERENCED AT : 910.000 912.000
RIGHT.HALFWORD	LABEL	REFERENCED AT : 3268.000 3270.000
RMEXP	REGISTER	REFERENCED AT : 6759.000 6820.000 6885.000 6942.000 6995.000 7051.000 7098.000 7143.000
RND	LABEL	REFERENCED AT : 4486.000 8380.000
RND	STRING	REFERENCED AT : 6767.000 6785.000 6828.000 6846.000 7008.000 7070.000
RO	STRING	REFERENCED AT : 3503.000 3730.000 3736.000 3744.000 3750.000 3757.000 3766.000 3783.000 3784.000 3803.000 3804.000 3822.000 3826.000 3911.000 4024.000 4140.000 4183.000 4464.000 4569.000 4571.000 4595.000 4614.000 4655.000 4816.000 4825.000 4833.000 5224.000 6875.000 6898.000 6902.000 6908.000 6917.000 6921.000 6927.000 6932.000 6955.000 6959.000 6965.000 6975.000 6979.000 6985.000 7089.000 7118.000 7122.000 7128.000 7133.000 7176.000 7181.000 7185.000
ROLL.OUT	LABEL	REFERENCED AT : 1108.000 1113.000
RP	REGISTER	REFERENCED AT : 6771.000 6788.000 6832.000 6849.000 7001.000 7002.000 7011.000 7065.000 7081.000 7104.000 7105.000 7157.000
RR1	LABEL	REFERENCED AT : 4514.000 8421.000 8424.000 8491.000 8492.000 8493.000 8496.000 8497.000
RRM	LABEL	REFERENCED AT : 4518.000 8429.000 8432.000 8500.000 8501.000
RRM1	LABEL	REFERENCED AT : 4507.000 8504.000 8505.000
RSCHNL	LABEL	REFERENCED AT : 7302.000
RSCHNL.WAIT.READY	LABEL	REFERENCED AT : 7658.000 7660.000 7662.000
RSCTL	LABEL	REFERENCED AT : 7311.000
RSTAEXP	STRING	REFERENCED AT : 916.000 1172.000 1735.000 1740.000
RSTCHNL.FLG	VARIABLE	REFERENCED AT : 542.000 7669.000
RSTFP	STRING	REFERENCED AT : 6730.000 6731.000 7026.000 7036.000 7170.000
RSTIPUFLG	STRING	REFERENCED AT : 980.000 986.000 7938.000
RSTPROTV	STRING	REFERENCED AT : 1172.000 1954.000 3126.000 7836.000 7842.000
RSTRHF	STRING	REFERENCED AT : 639.000 1135.000 1197.000 1233.000 1466.000 1473.000 1761.000 2490.000 3274.000 3442.000 3744.000 3757.000 4452.000 4453.000 4457.000 4470.000 4550.000 4630.000 4669.000 4962.000 4991.000 5013.000 5041.000 5077.000 5089.000 5096.000 5124.000 5154.000 5257.000 5268.000 5450.000 5545.000 6719.000 6869.000 7169.000
RSTX	LABEL	REFERENCED AT : 7505.000 7511.000
RSTX	STRING	REFERENCED AT : 1984.000 7396.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

RSTX.ACK	LABEL	REFERENCED AT : 824.000 839.000 843.000 849.000
RSTX.DRT	LABEL	REFERENCED AT : 7515.000 7523.000
RSTX.RETRY	LABEL	REFERENCED AT : 7513.000 7518.000
RSTX.TRANSFER.OK	LABEL	REFERENCED AT : 7533.000 7535.000
RTN.RHALF	LABEL	REFERENCED AT : 1089.000 1093.000
RTOM.ERROR	LABEL	REFERENCED AT : 1804.000 1809.000
RTOM.IOM.EXIT	LABEL	REFERENCED AT : 2687.000
RTOM.RSTX	LABEL	REFERENCED AT : 1799.000 1803.000 1813.000
RUN	STRING	REFERENCED AT : 983.000 1058.000 1127.000 1272.000 1536.000
RUN.HALT	LABEL	REFERENCED AT : 1044.000 1048.000 1054.000
RWCS	LABEL	REFERENCED AT : 5498.000 8386.000
RXE	STRING	REFERENCED AT : 7063.000 7065.000 7155.000 7157.000
RXXXXXXXX	LABEL	REFERENCED AT : 3709.000
RZ	VARIABLE	REFERENCED AT : 255.000
S	STRING	REFERENCED AT : 705.000 787.000 845.000 866.000 869.000 884.000 885.000 905.000 913.000 915.000 941.000 951.000 956.000 959.000 1026.000 1037.000 1044.000 1107.000 1112.000 1120.000 1231.000 1254.000 1263.000 1283.000 1319.000 1333.000 1374.000 1381.000 1385.000 1389.000 1398.000 1404.000 1405.000 1434.000 1450.000 1454.000 1556.000 1607.000 1642.000 1681.000 1684.000 1738.000 1740.000 1779.000 1818.000 1891.000 1897.000 1928.000 1930.000 1935.000 2039.000 2042.000 2045.000 2054.000 2086.000 2087.000 2118.000 2224.000 2243.000 2248.000 2296.000 2297.000 2310.000 2315.000 2325.000 2328.000 2340.000 2342.000 2370.000 2371.000 2462.000 2465.000 2471.000 2472.000 2474.000 2520.000 2551.000 2561.000 2562.000 2575.000 2586.000 2596.000 2607.000 2661.000 2668.000 2676.000 2678.000 2679.000 2684.000 2709.000 2719.000 2721.000 2789.000 2806.000 2808.000 2828.000 2835.000 2839.000 2842.000 2863.000 2871.000 2872.000 2888.000 2898.000 2916.000 2917.000 2918.000 2929.000 2931.000 2961.000 2965.000 2966.000 2970.000 3000.000 3003.000 3006.000 3008.000 3010.000 3027.000 3030.000 3033.000 3054.000 3057.000 3060.000 3063.000 3066.000 3081.000 3095.000 3098.000 3108.000 3110.000 3115.000 3117.000 3132.000 3135.000 3141.000 3155.000 3156.000 3162.000 3172.000 3180.000 3206.000 3213.000 3218.000 3241.000 3243.000 3305.000 3313.000 3316.000 3317.000 3320.000 3323.000 3324.000 3388.000 3390.000 3413.000 3418.000 3541.000 3564.000 3566.000 3570.000 3585.000 3597.000 3599.000 3611.000 3626.000 3627.000 3628.000 3629.000 3630.000 3631.000 3641.000 3645.000 3687.000 3691.000 3696.000 3705.000 3706.000 3708.000 3719.000 3731.000 3743.000 3767.000 3770.000 3772.000 3777.000 3790.000 3803.000 3823.000 3834.000 3835.000 3837.000 3857.000 3859.000 3876.000 3914.000 3937.000 3943.000 3962.000 3963.000 3965.000 3975.000 3993.000 3994.000 3996.000 3998.000 4000.000 4003.000 4010.000 4013.000 4028.000 4031.000 4041.000 4042.000 4066.000 4067.000 4075.000 4097.000 4107.000 4111.000 4118.000 4122.000 4131.000 4133.000 4145.000 4148.000 4149.000 4150.000 4151.000 4152.000 4234.000 4267.000 4270.000 4273.000 4276.000 4306.000 4368.000 4369.000 4370.000 4371.000 4372.000 4373.000 4374.000 4375.000 4377.000 4378.000 4405.000 4406.000 4457.000 4473.000 4492.000 4497.000 4498.000 4500.000 4511.000 4537.000 4537.000 4539.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

4544.000 4545.000 4545.000 4550.000 4560.000 4568.000 4583.000 4585.000 4586.000 4587.000
4589.000 4590.000 4590.000 4594.000 4602.000 4607.000 4611.000 4617.000 4618.000 4619.000
4649.000 4655.000 4670.000 4695.000 4704.000 4741.000 4760.000 4761.000 4763.000 4768.000
4777.000 4779.000 4861.000 4906.000 4909.000 4941.000 4964.000 4970.000 4993.000 4999.000
5023.000 5037.000 5048.000 5080.000 5111.000 5114.000 5125.000 5144.000 5154.000 5155.000
5158.000 5160.000 5174.000 5206.000 5219.000 5242.000 5246.000 5305.000 5309.000 5312.000
5362.000 5363.000 5411.000 5411.000 5431.000 5489.000 5499.000 5502.000 5506.000 5507.000
5510.000 5522.000 5532.000 5543.000 5546.000 5550.000 5553.000 5554.000 5574.000 5613.000
5618.000 5730.000 5759.000 5761.000 5777.000 5779.000 5781.000 5783.000 5785.000 5787.000
5791.000 5818.000 5826.000 5828.000 5833.000 5840.000 5850.000 5857.000 5868.000 5878.000
5930.000 5938.000 5940.000 5957.000 5958.000 5983.000 5995.000 6024.000 6027.000 6034.000
6139.000 6152.000 6157.000 6170.000 6181.000 6184.000 6185.000 6207.000 6219.000 6242.000
6243.000 6244.000 6245.000 6250.000 6324.000 6416.000 6464.000 6479.000 6501.000 6540.000
6543.000 6552.000 6564.000 6586.000 6594.000 6608.000 6628.000 6654.000 6654.000 6661.000
6662.000 6665.000 6671.000 6672.000 6676.000 6680.000 6682.000 6741.000 6807.000 6868.000
7207.000 7233.000 7246.000 7250.000 7275.000 7317.000 7342.000 7349.000 7354.000 7423.000
7424.000 7434.000 7466.000 7471.000 7530.000 7540.000 7599.000 7615.000 7630.000 7745.000
7758.000 7759.000 7800.000 7815.000 7831.000 7837.000 7896.000 7978.000 7982.000 7986.000
7993.000 8021.000 8030.000 8059.000 8384.000 8386.000 8387.000 8514.000 8515.000 8516.000
8517.000 8518.000 8519.000 8520.000 8521.000 8522.000 8523.000 8524.000 8525.000 8526.000
8527.000 8528.000 8529.000 8544.000

S REGISTER REFERENCED AT : 768.000 788.000 801.000 847.000 869.000 872.000 884.000 906.000
914.000 916.000 943.000 960.000 1036.000 1105.000 1107.000 1110.000 1112.000 1119.000
1120.000 1135.000 1154.000 1197.000 1231.000 1233.000 1235.000 1236.000 1254.000 1255.000
1265.000 1284.000 1285.000 1301.000 1302.000 1336.000 1382.000 1387.000 1391.000 1398.000
1399.000 1405.000 1408.000 1425.000 1458.000 1513.000 1516.000 1557.000 1576.000 1611.000
1644.000 1702.000 1739.000 1781.000 1812.000 1821.000 1891.000 1893.000 1895.000 1897.000
1898.000 1936.000 2041.000 2049.000 2055.000 2122.000 2140.000 2166.000 2167.000 2226.000
2229.000 2238.000 2239.000 2245.000 2249.000 2267.000 2299.000 2317.000 2331.000 2341.000
2371.000 2372.000 2466.000 2472.000 2474.000 2477.000 2536.000 2561.000 2578.000 2599.000
2610.000 2640.000 2661.000 2662.000 2665.000 2675.000 2678.000 2679.000 2686.000 2691.000
2695.000 2765.000 2828.000 2863.000 2872.000 2918.000 2931.000 2955.000 2969.000 3010.000
3066.000 3133.000 3153.000 3156.000 3167.000 3178.000 3204.000 3210.000 3215.000 3220.000
3242.000 3245.000 3306.000 3315.000 3319.000 3322.000 3324.000 3326.000 3347.000 3382.000
3424.000 3461.000 3479.000 3502.000 3503.000 3518.000 3529.000 3545.000 3550.000 3572.000
3574.000 3598.000 3633.000 3714.000 3763.000 3766.000 3767.000 3769.000 3771.000 3773.000
3777.000 3781.000 3794.000 3796.000 3797.000 3802.000 3805.000 3806.000 3808.000 3809.000
3816.000 3817.000 3818.000 3837.000 3846.000 3851.000 3879.000 3881.000 3889.000 3896.000
3904.000 3907.000 3917.000 3920.000 3921.000 3924.000 3925.000 3926.000 3930.000 3931.000
3943.000 3962.000 3965.000 3973.000 3977.000 3981.000 3985.000 3987.000 3992.000 3994.000
3995.000 3998.000 3999.000 4001.000 4007.000 4008.000 4010.000 4011.000 4012.000 4025.000
4027.000 4029.000 4046.000 4063.000 4064.000 4076.000 4080.000 4103.000 4138.000 4153.000
4192.000 4207.000 4221.000 4231.000 4235.000 4246.000 4250.000 4260.000 4288.000 4289.000
4292.000 4293.000 4294.000 4295.000 4296.000 4297.000 4298.000 4306.000 4379.000 4405.000
4472.000 4475.000 4476.000 4480.000 4482.000 4498.000 4500.000 4515.000 4530.000 4572.000
4576.000 4584.000 4587.000 4591.000 4596.000 4610.000 4615.000 4619.000 4661.000 4674.000
4687.000 4713.000 4744.000 4748.000 4761.000 4767.000 4785.000 4816.000 4843.000 4847.000
4870.000 4877.000 4895.000 4918.000 4945.000 4975.000 4981.000 4982.000 5004.000 5116.000
5133.000 5146.000 5173.000 5175.000 5177.000 5219.000 5224.000 5245.000 5308.000 5363.000
5367.000 5412.000 5431.000 5450.000 5451.000 5489.000 5503.000 5509.000 5544.000 5545.000
5547.000 5618.000 5731.000 5789.000 5793.000 5819.000 5827.000 5835.000 5851.000 5870.000
5887.000 5931.000 5940.000 5941.000 5984.000 5996.000 6025.000 6028.000 6037.000 6040.000
6076.000 6129.000 6155.000 6161.000 6171.000 6185.000 6209.000 6220.000 6252.000 6311.000
6325.000 6418.000 6465.000 6476.000 6480.000 6484.000 6516.000 6543.000 6545.000 6565.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

			6586.000 6590.000 6594.000 6599.000 6643.000 6658.000 6666.000 6671.000 6672.000 6674.000 6677.000 7210.000 7245.000 7275.000 7354.000 7467.000 7471.000 7538.000 7616.000 7638.000 7696.000 7731.000 7740.000 7746.000 7759.000 7784.000 7799.000 7816.000 7832.000 7839.000 7866.000 7958.000 7981.000 7984.000 7989.000 8031.000 8043.000 8059.000
S.P.ADDRESS.STOP	LABEL	REFERENCED AT :	5822.000 6331.000
S.P.ATTN	LABEL	REFERENCED AT :	750.000 1014.000 5733.000
S.P.BIBUSY	LABEL	REFERENCED AT :	5893.000 5896.000 7840.000
S.P.C.BYTE3	LABEL	REFERENCED AT :	5935.000 5937.000 6055.000 6080.000 6110.000 6134.000 6298.000 6329.000
S.P.CHECK.FUNCTION	LABEL	REFERENCED AT :	6029.000 6032.000
S.P.CHECK.MAP.ERROR	LABEL	REFERENCED AT :	5901.000 7826.000 7844.000
S.P.CLEAN.D	LABEL	REFERENCED AT :	5953.000 6056.000 6060.000 6082.000
S.P.CLEAR	LABEL	REFERENCED AT :	1024.000 5943.000 7810.000
S.P.CLEAR.ERROR	LABEL	REFERENCED AT :	5910.000 7850.000
S.P.CONVERT	LABEL	REFERENCED AT :	5972.000
S.P.CSW	LABEL	REFERENCED AT :	5968.000 6113.000
S.P.CSW.END	LABEL	REFERENCED AT :	6123.000 6131.000
S.P.CSW.READ	LABEL	REFERENCED AT :	6119.000
S.P.CSW.WRITE	LABEL	REFERENCED AT :	6118.000 6124.000
S.P.DATA	LABEL	REFERENCED AT :	5750.000 5796.000
S.P.DUD.CYCLE	LABEL	REFERENCED AT :	5887.000 6058.000
S.P.EFFECTIVE.ADDRESS	LABEL	REFERENCED AT :	5970.000 6005.000
S.P.ERROR.1	LABEL	REFERENCED AT :	5739.000 6362.000
S.P.ERROR.2	LABEL	REFERENCED AT :	5774.000 5804.000 6365.000
S.P.ERROR.3.4.5.7	LABEL	REFERENCED AT :	5886.000 5926.000 6039.000 6045.000 6121.000 6128.000 6210.000 6317.000 6370.000
S.P.ERROR.6	LABEL	REFERENCED AT :	5888.000 6042.000 6130.000 6367.000
S.P.ERROR.FUNNEL	LABEL	REFERENCED AT :	6358.000 6372.000
S.P.EVENREG	LABEL	REFERENCED AT :	6163.000
S.P.EXTENDED.1	LABEL	REFERENCED AT :	6195.000 6204.000
S.P.EXTFUNC	LABEL	REFERENCED AT :	5802.000 6265.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

S.P.EXTFUNC.1	LABEL	REFERENCED AT : 6269.000 6289.000
S.P.FAULT.2	LABEL	REFERENCED AT : 5729.000 5973.000 5992.000 6006.000 6088.000 6147.000 6214.000 6270.000 6307.000 6343.000
S.P.FAULT.3	LABEL	REFERENCED AT : 5984.000 5987.000 6068.000 6345.000
S.P.FAULT.4	LABEL	REFERENCED AT : 6010.000 6017.000 6347.000
S.P.FAULT.5	LABEL	REFERENCED AT : 6292.000 6349.000
S.P.FETCH.A	LABEL	REFERENCED AT : 5776.000 5918.000 5974.000 6295.000
S.P.FETCH.B	LABEL	REFERENCED AT : 5778.000 5883.000 6001.000 6069.000 6094.000 6125.000 6140.000 6224.000 6310.000 6431.000
S.P.FETCH.C	LABEL	REFERENCED AT : 5725.000 5780.000 5931.000 6013.000 6025.000 6048.000 6073.000 6105.000 6166.000 6178.000 6198.000 6227.000 6235.000 6257.000 6332.000 6352.000
S.P.FETCH.D	LABEL	REFERENCED AT : 5745.000 5782.000 6061.000 6169.000 6356.000 6371.000
S.P.FETCH.FUNCTION	LABEL	REFERENCED AT : 5784.000 5797.000 5981.000 5993.000 6007.000 6065.000 6089.000 6115.000 6150.000 6192.000 6205.000 6305.000
S.P.FETCH.TC.FLG.ID	LABEL	REFERENCED AT : 5786.000 5831.000 5837.000 5865.000 5875.000 5998.000
S.P.FLAG.ADDR	LABEL	REFERENCED AT : 6081.000 6111.000
S.P.FUNCTION.DATA	LABEL	REFERENCED AT : 5756.000
S.P.GPR	LABEL	REFERENCED AT : 5956.000 6146.000
S.P.GPR.READ	LABEL	REFERENCED AT : 6156.000
S.P.GPR.WRITE	LABEL	REFERENCED AT : 6154.000
S.P.INC.A	LABEL	REFERENCED AT : 5889.000 5917.000 5923.000
S.P.INC.PC	LABEL	REFERENCED AT : 5724.000 5890.000 5928.000
S.P.INC.RD	LABEL	REFERENCED AT : 5768.000 5922.000
S.P.INSERT.ARTR	LABEL	REFERENCED AT : 5834.000 5840.000
S.P.INSERT.CDTR	LABEL	REFERENCED AT : 5869.000 5879.000
S.P.INST.STEP	LABEL	REFERENCED AT : 5770.000 6213.000
S.P.INST.STOP	LABEL	REFERENCED AT : 5771.000 6226.000
S.P.INSTR.STEP	LABEL	REFERENCED AT : 1530.000 6216.000
S.P.IPL	LABEL	REFERENCED AT : 1273.000 6430.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

S.P.JUMP.TBL	LABEL	REFERENCED AT : 5759.000 5767.000
S.P.JUMP.TBL1	LABEL	REFERENCED AT : 5957.000 5964.000
S.P.KBDATA	LABEL	REFERENCED AT : 5799.000 6138.000
S.P.KEYBOARD	LABEL	REFERENCED AT : 5760.000 6191.000
S.P.KEYBOARD.OFF	LABEL	REFERENCED AT : 5751.000 5790.000 5800.000
S.P.M.INVALID	LABEL	REFERENCED AT : 7834.000 7841.000
S.P.M.PROTECT	LABEL	REFERENCED AT : 7835.000
S.P.MA	LABEL	REFERENCED AT : 6041.000 6047.000
S.P.MA.IN.A	LABEL	REFERENCED AT : 6028.000 6030.000
S.P.MAP.ERROR	LABEL	REFERENCED AT : 7827.000 7830.000
S.P.MEM.ADDR	LABEL	REFERENCED AT : 5965.000 6064.000
S.P.MEM.DATA	LABEL	REFERENCED AT : 5969.000 6023.000
S.P.MEMERR	LABEL	REFERENCED AT : 5902.000 7829.000
S.P.MEMORY.ADDRESS.TEST	LABEL	REFERENCED AT : 5882.000 5924.000 5929.000
S.P.MEMORY.READ	LABEL	REFERENCED AT : 5885.000 5894.000 5925.000 6038.000 6044.000 6120.000 6127.000 6209.000 6316.000
S.P.MEMORY.WRITE	LABEL	REFERENCED AT : 5884.000 5891.000 6037.000 6126.000
S.P.ODDREG	LABEL	REFERENCED AT : 6162.000 6175.000
S.P.OPRND.RD.STOP	LABEL	REFERENCED AT : 5772.000 6234.000
S.P.OPRND.WRT.STOP	LABEL	REFERENCED AT : 5773.000 6256.000
S.P.PC	LABEL	REFERENCED AT : 5967.000 6087.000
S.P.PEND.TEST	LABEL	REFERENCED AT : 5813.000
S.P.PSD2	LABEL	REFERENCED AT : 5971.000 5991.000
S.P.PSW	LABEL	REFERENCED AT : 5966.000 6304.000
S.P.PSW.RD	LABEL	REFERENCED AT : 1513.000 6312.000 7813.000
S.P.RD.OR.WT	LABEL	REFERENCED AT : 5805.000 5952.000
S.P.READ	LABEL	REFERENCED AT : 6035.000 6043.000
S.P.READ.PC	LABEL	REFERENCED AT : 6092.000 6100.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

S.P.READ.PSW	LABEL	REFERENCED AT :	1692.000 6308.000 6315.000
S.P.RESET.DISPA.PNDG	LABEL	REFERENCED AT :	5814.000 5830.000
S.P.RESET.DISPB.PNDG	LABEL	REFERENCED AT :	5815.000 5836.000
S.P.RESET.DISPC.PNDG	LABEL	REFERENCED AT :	5816.000 5864.000
S.P.RESET.DISPD.PNDG	LABEL	REFERENCED AT :	5817.000 5874.000
S.P.S.TO.FULLMAR	LABEL	REFERENCED AT :	5810.000 5825.000 5855.000
S.P.SAVE.FLAG	LABEL	REFERENCED AT :	6057.000 6285.000 6337.000
S.P.SET.A.C.D	LABEL	REFERENCED AT :	6083.000 6174.000 6279.000 6299.000
S.P.SET.ALL.FLAGS	LABEL	REFERENCED AT :	5728.000 5731.000 6137.000 6283.000 6330.000 6361.000
S.P.SET.B	LABEL	REFERENCED AT :	6144.000 6212.000 6277.000
S.P.SET.B.C.D	LABEL	REFERENCED AT :	5990.000 6021.000 6189.000 6202.000 6208.000 6281.000
S.P.SET.C.FLAG	LABEL	REFERENCED AT :	6218.000 6231.000 6239.000 6261.000
S.P.SET.ERR.9	LABEL	REFERENCED AT :	7836.000 7847.000
S.P.SET.ERR.A	LABEL	REFERENCED AT :	7842.000 7845.000
S.P.SET.PSW	LABEL	REFERENCED AT :	6320.000
S.P.TRNMT.COUNT.ADDR	LABEL	REFERENCED AT :	5835.000 5841.000 5873.000
S.P.UART.TBMT	LABEL	REFERENCED AT :	5736.000 5809.000 6246.000 6287.000
S.P.WRITE	LABEL	REFERENCED AT :	6036.000
S.P.WRITE.PC	LABEL	REFERENCED AT :	6093.000
S.P.WRITE.PSW	LABEL	REFERENCED AT :	6309.000
S.P.WRT.INC	LABEL	REFERENCED AT :	5769.000 5881.000
S.P.WRT.RD.STOP	LABEL	REFERENCED AT :	6232.000 6241.000 6254.000 6263.000
S.P.XMIT	LABEL	REFERENCED AT :	5812.000 5849.000
S.TO.T.DUD	LABEL	REFERENCED AT :	5777.000 5779.000 5781.000 5783.000 5785.000 5788.000 5870.000
SONETO	STRING	REFERENCED AT :	4600.000 4606.000
SAVE.BPIX.CPIX	LABEL	REFERENCED AT :	5358.000 5361.000
SAVE.DFFAULT.IPL	LABEL	REFERENCED AT :	1280.000
SAVE.DVC.ENTRY	LABEL	REFERENCED AT :	2679.000 2685.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

SAVE.IPL.ADDR	LABEL	REFERENCED AT : 1278.000 6433.000
SAVESIGN	STRING	REFERENCED AT : 1521.000 1550.000 1649.000 2266.000 3633.000 3680.000 3683.000 3762.000 3806.000 3822.000 3849.000 3898.000 3912.000 4080.000 4124.000 4184.000 4560.000 4568.000 7526.000
SBM	LABEL	REFERENCED AT : 4973.000 8297.000
SBRZBR	LABEL	REFERENCED AT : 4544.000 8250.000 8251.000
SCALE	LABEL	REFERENCED AT : 3612.000
SCALE	STRING	REFERENCED AT : 3611.000
SCALER	LABEL	REFERENCED AT : 3595.000 3605.000
SCHEDULE	LABEL	REFERENCED AT : 857.000
SCHEDULE.55.TRAP	LABEL	REFERENCED AT : 862.000 933.000 938.000
SCHEDULE.75.TRAP	LABEL	REFERENCED AT : 863.000 3232.000 7464.000
SCR.ADDR	VARIABLE	REFERENCED AT : 174.000 1197.000 1205.000 1208.000 1219.000 1224.000 1257.000 1258.000 1258.000 1466.000 1469.000
SCR.MASK	VARIABLE	REFERENCED AT : 300.000 1195.000 1203.000 1215.000 1255.000 1464.000
SCRATCH(	STRING	REFERENCED AT : 914.000 915.000 917.000 941.000 1107.000 1112.000 1119.000 1254.000 1257.000 1281.000 1283.000 1287.000 1289.000 1382.000 1391.000 1399.000 1408.000 1556.000 1625.000 1738.000 1774.000 1779.000 1785.000 1891.000 1897.000 1935.000 2224.000 2245.000 2249.000 2263.000 2275.000 2446.000 2501.000 2661.000 2684.000 3156.000 3213.000 3217.000 3241.000 3246.000 3313.000 3317.000 3324.000 5362.000 5368.000 5403.000 5411.000 5730.000 5747.000 5757.000 5777.000 5779.000 5781.000 5783.000 5785.000 5787.000 5791.000 5794.000 5818.000 5821.000 5826.000 5833.000 5840.000 5842.000 5845.000 5850.000 5858.000 5860.000 5868.000 5872.000 5878.000 5914.000 5920.000 5927.000 5930.000 5938.000 5942.000 5945.000 5979.000 5983.000 5989.000 5995.000 5997.000 6000.000 6012.000 6018.000 6020.000 6024.000 6027.000 6034.000 6046.000 6063.000 6072.000 6104.000 6122.000 6132.000 6136.000 6143.000 6152.000 6164.000 6168.000 6173.000 6176.000 6180.000 6181.000 6188.000 6196.000 6197.000 6201.000 6207.000 6211.000 6219.000 6221.000 6223.000 6249.000 6250.000 6253.000 6272.000 6273.000 6275.000 6276.000 6286.000 6294.000 6296.000 6313.000 6318.000 6321.000 6323.000 6324.000 6335.000 6351.000 6355.000 6360.000 6369.000 6550.000 6564.000 7352.000 7462.000 7466.000 7483.000 7540.000 7555.000 7615.000 7622.000 7627.000 7641.000 7684.000 7752.000 7805.000 7815.000 7818.000 7831.000 7837.000 7978.000 7988.000 8030.000 8037.000
SCT.FLG	VARIABLE	REFERENCED AT : 557.000 594.000 7279.000 7441.000 7509.000 7519.000 7525.000 7534.000 7569.000 7582.000 7605.000
SCZ	LABEL	REFERENCED AT : 4759.000 8120.000 8198.000
SCZ2	LABEL	REFERENCED AT : 4762.000 4763.000 4765.000
SDEST	STRING	REFERENCED AT : 594.000 872.000 984.000 1111.000 1124.000 1205.000 1208.000 1219.000 1224.000 1258.000 1279.000 1464.000 1469.000 1701.000 1811.000 1814.000 1866.000 1890,000 1896.000 2264.000 2435.000 2622.000 2660.000 2667.000 2764.000 2886.000 3472.000 3633.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                3651.000 3680.000 3683.000 3711.000 3794.000 3806.000 3878.000 3880.000 3889.000 3906.000
                                3924.000 3988.000 4000.000 4055.000 4073.000 4078.000 4124.000 4193.000 4209.000 4246.000
                                4260.000 5077.000 5547.000 5944.000 6114.000 7279.000 7441.000 7496.000 7502.000 7509.000
                                7519.000 7525.000 7534.000 7569.000 7574.000 7582.000 7605.000 7669.000 7674.000 7677.000
                                7821.000 7860.000 7885.000 7913.000 8049.000

SE                      STRING  REFERENCED AT :  2141.000 2266.000 2362.000 3738.000 3745.000 3752.000 3758.000 4553.000
                                4555.000 4661.000 4731.000 4752.000 4952.000

SEA                     LABEL   REFERENCED AT :  5047.000 8388.000

SELSPARE                STRING  REFERENCED AT :  3988.000

SERIAL.PANEL            STRING  REFERENCED AT :  1013.000 1023.000 1272.000 1512.000 1578.000 1618.000 1690.000 1701.000
                                1712.000 1777.000 5899.000 7809.000

SET(                    STRING  REFERENCED AT :   681.000  786.000  787.000  916.000  954.000  965.000  981.000  982.000
                                1029.000 1030.000 1035.000 1057.000 1081.000 1101.000 1126.000 1128.000 1166.000 1170.000
                                1171.000 1236.000 1270.000 1271.000 1288.000 1293.000 1471.000 1497.000 1535.000 1536.000
                                1577.000 1680.000 1688.000 1713.000 1724.000 1745.000 1788.000 1841.000 1953.000 2229.000
                                2524.000 2563.000 2969.000 3101.000 3125.000 3139.000 3163.000 3171.000 3212.000 3323.000
                                3348.000 3386.000 3418.000 3488.000 3516.000 3769.000 3796.000 4389.000 4398.000 4459.000
                                4470.000 5173.000 5196.000 5239.000 5279.000 5313.000 5316.000 5346.000 5353.000 5371.000
                                5390.000 5459.000 5463.000 5500.000 5503.000 5526.000 5545.000 5592.000 5593.000 5595.000
                                5734.000 5748.000 5846.000 5857.000 5948.000 5976.000 6243.000 6363.000 6384.000 6541.000
                                6590.000 6640.000 6660.000 6663.000 6807.000 6875.000 6932.000 7089.000 7133.000 7458.000
                                7640.000 7697.000 7761.000 7784.000 7799.000 7812.000 7832.000 7858.000 7928.000 8014.000
                                8017.000 8024.000 8042.000

SET.55.EXIT             LABEL   REFERENCED AT :  7892.000 8010.000 8018.000

SET.75.M                LABEL   REFERENCED AT :  7959.000 7964.000

SET.AEXP                LABEL   REFERENCED AT :  3565.000 3570.000 3790.000

SET.BLOCKED.M.FLAG      LABEL   REFERENCED AT :   802.000 3219.000 3426.000 8040.000

SET.FLSA.M              LABEL   REFERENCED AT :  7961.000 7968.000

SET.ENBL.AEXP           LABEL   REFERENCED AT :  7960.000 7966.000

SET.MAP.N.FOUND         LABEL   REFERENCED AT :  6535.000 6551.000

SET.OTHERS              LABEL   REFERENCED AT :  5275.000 5277.000 5278.000

SET.PSD1.AEXP           LABEL   REFERENCED AT :   884.000 1481.000 1483.000

SET.PSW.STATUS          LABEL   REFERENCED AT :  1683.000

SET.RHFLAG.IND          LABEL   REFERENCED AT :  3272.000

SET.SREG.0              LABEL   REFERENCED AT :  3637.000 3639.000

SET.SYSTEM.CONSTANTS    LABEL   REFERENCED AT :  1164.000 1268.000 1327.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 7 2 7 5 C P U

SETAEXP	STRING	REFERENCED AT : 3567.000
SETCAR	STRING	REFERENCED AT : 3503.000 3766.000 3836.000 3905.000 3920.000 3930.000 3942.000 3961.000 3964.000 4002.000 4007.000 4077.000 4084.000 4091.000 4229.000 4244.000 4256.000 4272.000 4301.000 4305.000 4816.000 4825.000
SETCC(	STRING	REFERENCED AT : 1231.000 2961.000 3107.000 3503.000 3527.000 3566.000 3572.000 3574.000 3648.000 3785.000 3788.000 3826.000 4140.000 4492.000 4507.000 4530.000 4539.000 4544.000 4555.000 4576.000 4708.000 4712.000 4715.000 4727.000 4731.000 4733.000 4744.000 4748.000 4761.000 4816.000 4825.000 4941.000 4977.000 5004.000 5080.000 5219.000 5554.000 7074.000 7342.000 7354.000
SETCC(#)	STRING	REFERENCED AT : 3529.000 4476.000 4480.000 4482.000 4487.000 4488.000 4498.000 4508.000 4514.000 4515.000 4517.000 4519.000 4520.000 4531.000 4537.000 4553.000 4657.000 4661.000 4674.000 4687.000 4950.000 4952.000 4982.000 5537.000 6771.000 6774.000 6779.000 6788.000 6791.000 6796.000 6832.000 6835.000 6840.000 6849.000 6852.000 6857.000 6897.000 6901.000 6907.000 6916.000 6920.000 6926.000 6954.000 6958.000 6964.000 6974.000 6978.000 6984.000 7011.000 7014.000 7019.000 7078.000 7081.000 7084.000 7117.000 7121.000 7127.000 7175.000 7180.000 7184.000 7993.000
SETCC.EXIT	LABEL	REFERENCED AT : 7353.000 7357.000
SETCPU	LABEL	REFERENCED AT : 8006.000 8430.000
SETCPU.EXIT	LABEL	REFERENCED AT : 8016.000 8019.000 8027.000
SETRHF	STRING	REFERENCED AT : 1094.000 1218.000 1467.000 2164.000 4475.000 5079.000 5095.000 5139.000 5170.000 5197.000 5326.000 5536.000
SETXCC(	STRING	REFERENCED AT : 885.000 956.000 1684.000 2562.000 4406.000 5312.000 6662.000 6682.000
SHIFT(	STRING	REFERENCED AT : 6753.000 6756.000 6762.000 6767.000 6771.000 6774.000 6777.000 6779.000 6782.000 6785.000 6788.000 6791.000 6794.000 6796.000 6814.000 6817.000 6823.000 6828.000 6832.000 6835.000 6838.000 6840.000 6843.000 6846.000 6849.000 6852.000 6855.000 6857.000 6877.000 6882.000 6888.000 6893.000 6897.000 6898.000 6901.000 6902.000 6905.000 6907.000 6908.000 6913.000 6916.000 6917.000 6920.000 6921.000 6924.000 6926.000 6927.000 6934.000 6939.000 6945.000 6950.000 6954.000 6955.000 6958.000 6959.000 6962.000 6964.000 6965.000 6971.000 6974.000 6975.000 6978.000 6979.000 6982.000 6984.000 6985.000 6990.000 6995.000 6998.000 7004.000 7005.000 7008.000 7011.000 7014.000 7017.000 7018.000 7019.000 7046.000 7051.000 7054.000 7055.000 7057.000 7061.000 7063.000 7066.000 7070.000 7076.000 7078.000 7081.000 7083.000 7084.000 7091.000 7098.000 7101.000 7110.000 7114.000 7117.000 7118.000 7121.000 7122.000 7125.000 7127.000 7128.000 7135.000 7136.000 7143.000 7146.000 7147.000 7149.000 7153.000 7155.000 7158.000 7162.000 7173.000 7175.000 7176.000 7180.000 7181.000 7183.000 7184.000 7185.000
SHIFTD	LABEL	REFERENCED AT : 4566.000 8124.000 8125.000 8202.000 8203.000
SHIFTD(	STRING	REFERENCED AT : 2312.000 2314.000 3941.000 4300.000 4304.000 4570.000
SHIFTDI(	STRING	REFERENCED AT : 2313.000 3770.000 3772.000 3773.000 3876.000 3963.000
SHIFTIO	STRING	REFERENCED AT : 1096.000 4500.000 4525.000 4540.000 4545.000 4573.000 5098.000 5141.000 5177.000 5201.000 5329.000
SHIFTS	LABEL	REFERENCED AT : 4558.000 8121.000 8122.000 8123.000 8199.000 8200.000 8201.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

SHIFTS(	STRING	REFERENCED AT :	1931.000 2047.000 2048.000 2890.000 2900.000 3414.000 4287.000 4562.000 6656.000 6673.000 7234.000 7239.000 7531.000
SHIFTX1	LABEL	REFERENCED AT :	4572.000 8595.000 8596.000 8597.000 8598.000 8599.000 8604.000 8605.000 8607.000
SHIFTX2	LABEL	REFERENCED AT :	4576.000 8603.000 8606.000
SI.VECTOR	VARIABLE	REFERENCED AT :	247.000 870.000 937.000 960.000 1906.000 2690.000 2696.000 3226.000 7527.000 7596.000
SIGNSAVE	STRING	REFERENCED AT :	1524.000 1554.000 1653.000 2269.000 3717.000 3783.000 3825.000 3879.000 3904.000 4001.000 4267.000 4577.000 7529.000
SIGNSAVE7	STRING	REFERENCED AT :	2965.000
SINGLE.CTL.IOCD	LABEL	REFERENCED AT :	2364.000 2376.000
SINGLE.IOCD	LABEL	REFERENCED AT :	2368.000
SIO	LABEL	REFERENCED AT :	7293.000
SIPU	LABEL	REFERENCED AT :	1491.000 8385.000
SIPU.IPU.TRAP	LABEL	REFERENCED AT :	1492.000 7853.000
SIPU.RETURN	LABEL	REFERENCED AT :	921.000 1494.000
SLA	STRING	REFERENCED AT :	6753.000 6762.000 6767.000 6774.000 6777.000 6782.000 6785.000 6791.000 6794.000 6814.000 6823.000 6828.000 6835.000 6838.000 6843.000 6846.000 6852.000 6855.000 6877.000 6888.000 6893.000 6901.000 6905.000 6913.000 6920.000 6924.000 6934.000 6945.000 6950.000 6958.000 6962.000 6971.000 6978.000 6982.000 6990.000 6995.000 7004.000 7005.000 7008.000 7014.000 7017.000 7018.000 7046.000 7055.000 7057.000 7061.000 7066.000 7070.000 7076.000 7083.000 7084.000 7091.000 7098.000 7110.000 7114.000 7121.000 7125.000 7136.000 7147.000 7149.000 7153.000 7158.000 7162.000 7173.000 7183.000 7184.000 7185.000 8603.000
SLAD	STRING	REFERENCED AT :	8606.000
SLC	STRING	REFERENCED AT :	2047.000 2048.000 6673.000 8605.000
SLCD	STRING	REFERENCED AT :	3770.000 3772.000 3773.000 3876.000
SLFFT	REGISTER	REFERENCED AT :	1046.000 1047.000 1930.000 1934.000 3417.000 3770.000 3772.000 3859.000 3876.000 3963.000 4459.000 4763.000 4778.000 4907.000 5156.000 5160.000 6534.000
SLL	STRING	REFERENCED AT :	1931.000 3414.000 6898.000 6902.000 6906.000 6917.000 6921.000 6927.000 6955.000 6959.000 6965.000 6975.000 6979.000 6985.000 7065.000 7118.000 7122.000 7128.000 7157.000 7176.000 7181.000 8604.000
SLLD	STRING	REFERENCED AT :	2312.000 3941.000 3963.000 4300.000 4304.000 8607.000
SNIBL	REGISTER	REFERENCED AT :	410.000 1434.000 1435.000 2311.000 2887.000 3413.000 3687.000 3691.000 3696.000 3708.000 3834.000 3835.000 4097.000 4107.000 4111.000 4118.000 4131.000 4145.000 4148.000 4149.000 4150.000 4151.000 4152.000 4208.000 4210.000 4219.000 4222.000 4248.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

```
                                4251.000 4276.000 4378.000 4589.000 4611.000 4617.000 4618.000 5761.000 5857.000 5958.000
                                6115.000 6140.000 6142.000 6182.000 7250.000

SNIBR                  REGISTER REFERENCED AT :  1037.000 2045.000 2087.000 2089.000 2965.000 2966.000 3626.000 3627.000
                                3628.000 3629.000 3630.000 3631.000 3644.000 3646.000 3705.000 3711.000 3720.000 3823.000
                                3914.000 3927.000 4013.000 4042.000 4067.000 4122.000 4133.000 4263.000 4368.000 4369.000
                                4370.000 4371.000 4372.000 4373.000 4374.000 4375.000 4377.000 4457.000 5026.000 5038.000
                                5155.000 5811.000 5828.000 6186.000 7424.000 7426.000

SPSHIFT(               STRING   REFERENCED AT :  6999.000 7001.000 7065.000 7102.000 7104.000 7157.000

SRA                    STRING   REFERENCED AT :  6771.000 6779.000 6788.000 6796.000 6832.000 6840.000 6849.000 6857.000
                                6897.000 6907.000 6916.000 6926.000 6954.000 6964.000 6974.000 6984.000 6998.000 6999.000
                                7001.000 7011.000 7019.000 7051.000 7054.000 7063.000 7078.000 7081.000 7101.000 7102.000
                                7104.000 7117.000 7127.000 7135.000 7143.000 7146.000 7155.000 7175.000 7180.000 8595.000

SRAD                   STRING   REFERENCED AT :  8598.000

SRC                    STRING   REFERENCED AT :  8597.000

SRL                    STRING   REFERENCED AT :  2313.000 4287.000 6656.000 7234.000 7239.000 7531.000 8577.000 8578.000
                                8596.000

SRLD                   STRING   REFERENCED AT :  2314.000 8599.000

ST                     LABEL    REFERENCED AT :  4844.000 4850.000 4891.000 8319.000 8320.000

ST.DWORD               LABEL    REFERENCED AT :  4893.000

START.IOC              LABEL    REFERENCED AT :  2382.000 2399.000

STATUS                 REGISTER REFERENCED AT :   786.000 1570.000 1608.000 3171.000 3224.000 3389.000 3442.000 3531.000
                                3554.000 3557.000 3790.000 5039.000 5052.000 5111.000 5154.000 6442.000 6660.000 6668.000
                                6675.000

STATUS.CC              VARIABLE REFERENCED AT :   226.000 7349.000 7538.000

STEP.HALT              LABEL    REFERENCED AT :  1052.000 6338.000

STF                    LABEL    REFERENCED AT :  4905.000 8321.000

STF.EXIT               LABEL    REFERENCED AT :  4907.000 4927.000

STF1                   LABEL    REFERENCED AT :  4909.000 4913.000

STM                    LABEL    REFERENCED AT :  4860.000 8149.000

STM.DWORD              LABEL    REFERENCED AT :  4894.000

STMASK                 VARIABLE REFERENCED AT :   155.000  785.000  869.000 1169.000 1606.000 1681.000 2551.000 3170.000
                                3221.000 3351.000 3427.000 5111.000 5114.000

STOP                   LABEL    REFERENCED AT :   677.000 1008.000 2099.000

STORE.CNT              LABEL    REFERENCED AT :  2462.000 2472.000 2475.000
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

```
STORE.CPU.STATUS.OR.NUM          LABEL   REFERENCED AT :   958.000

STORE.DOUBLE.INDIR               LABEL   REFERENCED AT :  4873.000 5221.000

STORE.DOUBLE.INDIR.RTN           LABEL   REFERENCED AT :  4878.000 5225.000

STORE.EXIT                       LABEL   REFERENCED AT :  4892.000 4898.000 4946.000 4953.000

STORE.TOCDO                      LABEL   REFERENCED AT :  2372.000 2380.000

STORE.PSD2                       LABEL   REFERENCED AT :  5348.000 5370.000 5373.000

STORE.PSW1                       LABEL   REFERENCED AT :   872.000  883.000

STORE.STATUS                     LABEL   REFERENCED AT :   962.000

STORE.STATUS.EXIT                LABEL   REFERENCED AT :   964.000  967.000

STPIO                            LABEL   REFERENCED AT :  7299.000

SUF                              LABEL   REFERENCED AT :  3581.000 8352.000

SUFD                             LABEL   REFERENCED AT :  4023.000

SUFW                             LABEL   REFERENCED AT :  3589.000

SVC                              LABEL   REFERENCED AT :  3405.000 8404.000

SVC.1                            LABEL   REFERENCED AT :  3411.000 7903.000

SVC.CHECK                        LABEL   REFERENCED AT :  3407.000 7894.000

SVC.ERROR                        LABEL   REFERENCED AT :  3420.000

SVC.TRAP.ENTRY.POINT             LABEL   REFERENCED AT :  3223.000 3427.000 7931.000

SWAP                             LABEL   REFERENCED AT :  4508.000 4514.000 4515.000 4517.000 4519.000 4520.000 4525.000

SWAP2                            LABEL   REFERENCED AT :  4537.000 4540.000

SYSTEM.CHECK.STATUS              LABEL   REFERENCED AT :  3203.000 3359.000 7471.000

SYSTEM.CHECK.TRAP                LABEL   REFERENCED AT :  2437.000 3201.000 6553.000 7275.000 7345.000 7435.000 7670.000

SYSTEM.RESET                     LABEL   REFERENCED AT :  1020.000 1149.000 7806.000

T                                STRING  REFERENCED AT :   639.000  676.000  677.000  678.000  682.000  785.000  786.000  817.000
                                          859.000  864.000  868.000  872.000  897.000  909.000  910.000  911.000  916.000  928.000
                                          936.000  943.000  948.000  949.000  961.000  969.000  970.000  984.000  996.000 1004.000
                                         1015.000 1027.000 1056.000 1064.000 1086.000 1088.000 1092.000 1110.000 1121.000 1165.000
                                         1167.000 1193.000 1194.000 1230.000 1255.000 1265.000 1274.000 1276.000 1277.000 1286.000
                                         1288.000 1294.000 1297.000 1298.000 1310.000 1332.000 1336.000 1338.000 1340.000 1345.000
                                         1361.000 1363.000 1375.000 1380.000 1384.000 1386.000 1406.000 1407.000 1433.000 1456.000
                                         1478.000 1479.000 1484.000 1513.000 1516.000 1520.000 1532.000 1557.000 1570.000 1573.000
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

1574.000 1576.000 1583.000 1585.000 1594.000 1606.000 1608.000 1609.000 1611.000 1612.000
1624.000 1644.000 1656.000 1701.000 1702.000 1704.000 1705.000 1715.000 1726.000 1731.000
1734.000 1767.000 1772.000 1773.000 1780.000 1781.000 1784.000 1796.000 1811.000 1813.000
1814.000 1821.000 1842.000 1849.000 1859.000 1863.000 1865.000 1866.000 1904.000 1929.000
1934.000 2041.000 2053.000 2070.000 2089.000 2090.000 2121.000 2122.000 2140.000 2184.000
2192.000 2229.000 2239.000 2262.000 2264.000 2267.000 2272.000 2274.000 2290.000 2309.000
2324.000 2363.000 2369.000 2372.000 2377.000 2378.000 2379.000 2394.000 2397.000 2401.000
2402.000 2424.000 2435.000 2447.000 2455.000 2466.000 2468.000 2473.000 2477.000 2497.000
2498.000 2524.000 2543.000 2560.000 2578.000 2599.000 2610.000 2622.000 2640.000 2665.000
2672.000 2674.000 2677.000 2688.000 2710.000 2713.000 2728.000 2737.000 2765.000 2767.000
2769.000 2824.000 2826.000 2859.000 2861.000 2870.000 2897.000 2915.000 2927.000 2930.000
2939.000 2945.000 2955.000 2969.000 2997.000 3025.000 3034.000 3065.000 3102.000 3121.000
3126.000 3133.000 3134.000 3147.000 3153.000 3164.000 3166.000 3170.000 3174.000 3178.000
3179.000 3204.000 3205.000 3210.000 3216.000 3221.000 3242.000 3245.000 3306.000 3309.000
3315.000 3319.000 3322.000 3326.000 3347.000 3351.000 3354.000 3387.000 3389.000 3411.000
3414.000 3415.000 3422.000 3427.000 3442.000 3451.000 3452.000 3461.000 3463.000 3464.000
3516.000 3529.000 3531.000 3540.000 3543.000 3545.000 3546.000 3550.000 3551.000 3554.000
3557.000 3563.000 3582.000 3583.000 3587.000 3610.000 3635.000 3643.000 3651.000 3671.000
3690.000 3694.000 3699.000 3702.000 3711.000 3722.000 3730.000 3736.000 3744.000 3750.000
3757.000 3765.000 3768.000 3788.000 3789.000 3812.000 3814.000 3821.000 3836.000 3855.000
3878.000 3880.000 3893.000 3896.000 3906.000 3907.000 3911.000 3918.000 3920.000 3935.000
3990.000 3992.000 3995.000 4002.000 4007.000 4029.000 4040.000 4050.000 4055.000 4065.000
4073.000 4077.000 4147.000 4183.000 4196.000 4209.000 4215.000 4221.000 4226.000 4240.000
4244.000 4250.000 4255.000 4258.000 4269.000 4272.000 4278.000 4301.000 4302.000 4305.000
4404.000 4422.000 4429.000 4459.000 4464.000 4468.000 4472.000 4486.000 4507.000 4511.000
4516.000 4517.000 4518.000 4530.000 4534.000 4576.000 4595.000 4608.000 4616.000 4712.000
4746.000 4752.000 4762.000 4764.000 4785.000 4790.000 4814.000 4823.000 4825.000 4831.000
4833.000 4843.000 4847.000 4870.000 4877.000 4895.000 4912.000 4945.000 4950.000 4952.000
4975.000 4981.000 5039.000 5052.000 5112.000 5117.000 5156.000 5224.000 5243.000 5244.000
5275.000 5277.000 5317.000 5319.000 5323.000 5330.000 5354.000 5360.000 5364.000 5365.000
5366.000 5367.000 5451.000 5499.000 5503.000 5509.000 5521.000 5547.000 5566.000 5571.000
5727.000 5746.000 5749.000 5789.000 5811.000 5819.000 5820.000 5832.000 5838.000 5839.000
5843.000 5844.000 5856.000 5866.000 5871.000 5876.000 5899.000 5904.000 5906.000 5907.000
5913.000 5915.000 5919.000 5933.000 5936.000 5941.000 5944.000 5955.000 5978.000 5984.000
5986.000 5988.000 5996.000 5999.000 6014.000 6016.000 6019.000 6028.000 6037.000 6050.000
6052.000 6054.000 6056.000 6062.000 6070.000 6071.000 6074.000 6076.000 6078.000 6079.000
6092.000 6099.000 6103.000 6106.000 6107.000 6109.000 6114.000 6129.000 6133.000 6135.000
6141.000 6142.000 6148.000 6159.000 6161.000 6167.000 6171.000 6172.000 6179.000 6182.000
6186.000 6187.000 6194.000 6199.000 6200.000 6220.000 6222.000 6225.000 6229.000 6232.000
6237.000 6240.000 6251.000 6252.000 6259.000 6262.000 6266.000 6271.000 6274.000 6278.000
6280.000 6282.000 6284.000 6293.000 6297.000 6306.000 6322.000 6325.000 6327.000 6333.000
6334.000 6336.000 6344.000 6346.000 6348.000 6350.000 6353.000 6354.000 6359.000 6364.000
6366.000 6368.000 6372.000 6382.000 6390.000 6465.000 6480.000 6485.000 6514.000 6534.000
6548.000 6609.000 6629.000 6634.000 6643.000 6663.000 6668.000 6675.000 6677.000 6695.000
6702.000 7205.000 7206.000 7209.000 7210.000 7243.000 7273.000 7274.000 7340.000 7348.000
7351.000 7426.000 7428.000 7451.000 7461.000 7467.000 7470.000 7481.000 7484.000 7527.000
7529.000 7536.000 7553.000 7556.000 7596.000 7600.000 7616.000 7626.000 7638.000 7652.000
7658.000 7666.000 7668.000 7672.000 7682.000 7687.000 7689.000 7696.000 7725.000 7731.000
7740.000 7744.000 7747.000 7756.000 7811.000 7817.000 7820.000 7821.000 7832.000 7839.000
7846.000 7848.000 7856.000 7859.000 7860.000 7865.000 7866.000 7879.000 7881.000 7882.000
7884.000 7885.000 7901.000 7910.000 7912.000 7913.000 7955.000 7958.000 7959.000 7963.000
7965.000 7967.000 7969.000 7981.000 7984.000 7987.000 7989.000 7990.000 7992.000 8031.000
8033.000 8034.000 8035.000 8043.000 8048.000 8056.000 8061.000

T

REGISTER REFERENCED AT : 692.000 786.000 800.000 821.000 822.000 835.000 846.000 860.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

868.000	871.000	872.000	884.000	906.000	910.000	911.000	914.000	917.000	937.000
946.000	948.000	949.000	984.000	997.000	1002.000	1009.000	1011.000	1012.000	1013.000
1016.000	1024.000	1028.000	1044.000	1056.000	1059.000	1087.000	1088.000	1089.000	1101.000
1120.000	1121.000	1128.000	1167.000	1168.000	1193.000	1194.000	1195.000	1220.000	1222.000
1232.000	1257.000	1265.000	1268.000	1270.000	1275.000	1279.000	1281.000	1287.000	1288.000
1289.000	1298.000	1299.000	1300.000	1309.000	1327.000	1362.000	1365.000	1366.000	1367.000
1381.000	1385.000	1386.000	1387.000	1407.000	1408.000	1426.000	1427.000	1436.000	1458.000
1466.000	1478.000	1479.000	1484.000	1521.000	1533.000	1550.000	1551.000	1571.000	1574.000
1576.000	1608.000	1609.000	1611.000	1612.000	1625.000	1649.000	1650.000	1702.000	1705.000
1716.000	1724.000	1746.000	1768.000	1773.000	1774.000	1775.000	1781.000	1783.000	1785.000
1815.000	1816.000	1817.000	1838.000	1839.000	1844.000	1863.000	1866.000	1890.000	1894.000
1896.000	1899.000	1901.000	1905.000	1906.000	1911.000	1931.000	1932.000	1936.000	1940.000
2043.000	2054.000	2066.000	2067.000	2077.000	2090.000	2092.000	2117.000	2118.000	2124.000
2127.000	2142.000	2143.000	2152.000	2153.000	2181.000	2183.000	2193.000	2201.000	2206.000
2217.000	2222.000	2223.000	2235.000	2245.000	2256.000	2257.000	2263.000	2264.000	2265.000
2275.000	2291.000	2292.000	2298.000	2316.000	2327.000	2364.000	2379.000	2402.000	2422.000
2446.000	2467.000	2468.000	2470.000	2474.000	2496.000	2498.000	2499.000	2501.000	2516.000
2527.000	2548.000	2561.000	2573.000	2584.000	2593.000	2605.000	2618.000	2660.000	2663.000
2664.000	2666.000	2667.000	2668.000	2672.000	2674.000	2675.000	2679.000	2682.000	2689.000
2690.000	2696.000	2709.000	2711.000	2712.000	2767.000	2768.000	2772.000	2789.000	2805.000
2825.000	2828.000	2834.000	2838.000	2860.000	2863.000	2871.000	2905.000	2918.000	2930.000
2931.000	2942.000	2956.000	2967.000	2970.000	2971.000	2981.000	3010.000	3066.000	3135.000
3152.000	3159.000	3160.000	3167.000	3171.000	3180.000	3206.000	3211.000	3216.000	3217.000
3224.000	3229.000	3242.000	3243.000	3245.000	3246.000	3309.000	3350.000	3357.000	3388.000
3415.000	3418.000	3421.000	3423.000	3443.000	3452.000	3464.000	3476.000	3479.000	3503.000
3518.000	3522.000	3527.000	3541.000	3545.000	3546.000	3547.000	3548.000	3550.000	3551.000
3563.000	3564.000	3570.000	3582.000	3584.000	3585.000	3587.000	3595.000	3633.000	3643.000
3645.000	3670.000	3671.000	3706.000	3722.000	3743.000	3767.000	3769.000	3770.000	3772.000
3777.000	3782.000	3785.000	3790.000	3794.000	3795.000	3796.000	3797.000	3806.000	3807.000
3808.000	3809.000	3812.000	3813.000	3816.000	3817.000	3818.000	3826.000	3836.000	3841.000
3846.000	3859.000	3876.000	3880.000	3889.000	3894.000	3897.000	3901.000	3905.000	3915.000
3920.000	3925.000	3926.000	3929.000	3930.000	3938.000	3991.000	3995.000	4002.000	4007.000
4011.000	4012.000	4027.000	4028.000	4035.000	4036.000	4038.000	4045.000	4060.000	4061.000
4077.000	4084.000	4091.000	4092.000	4124.000	4140.000	4190.000	4205.000	4216.000	4217.000
4228.000	4229.000	4242.000	4244.000	4245.000	4257.000	4258.000	4262.000	4269.000	4272.000
4278.000	4284.000	4285.000	4292.000	4293.000	4294.000	4295.000	4296.000	4297.000	4298.000
4404.000	4405.000	4471.000	4473.000	4488.000	4493.000	4499.000	4507.000	4508.000	4514.000
4515.000	4516.000	4517.000	4518.000	4519.000	4520.000	4530.000	4531.000	4537.000	4577.000
4578.000	4597.000	4607.000	4614.000	4637.000	4713.000	4744.000	4748.000	4764.000	4766.000
4781.000	4789.000	4790.000	4816.000	4818.000	4825.000	4827.000	4833.000	4835.000	4849.000
4871.000	4880.000	4881.000	4897.000	4922.000	4944.000	4969.000	4998.000	5040.000	5112.000
5116.000	5117.000	5134.000	5157.000	5194.000	5196.000	5211.000	5247.000	5255.000	5257.000
5280.000	5308.000	5309.000	5311.000	5314.000	5319.000	5321.000	5324.000	5343.000	5345.000
5346.000	5347.000	5349.000	5351.000	5353.000	5360.000	5363.000	5365.000	5366.000	5367.000
5368.000	5403.000	5452.000	5455.000	5501.000	5510.000	5532.000	5544.000	5545.000	5546.000
5547.000	5574.000	5593.000	5594.000	5726.000	5731.000	5746.000	5747.000	5757.000	5759.000
5798.000	5801.000	5803.000	5812.000	5814.000	5815.000	5816.000	5820.000	5821.000	5832.000
5838.000	5839.000	5843.000	5845.000	5860.000	5866.000	5867.000	5872.000	5876.000	5877.000
5901.000	5911.000	5914.000	5916.000	5919.000	5920.000	5932.000	5941.000	5942.000	5945.000
5946.000	5949.000	5975.000	5979.000	5982.000	5985.000	5986.000	5988.000	5989.000	5994.000
5997.000	5999.000	6000.000	6003.000	6008.000	6012.000	6014.000	6015.000	6016.000	6018.000
6020.000	6026.000	6033.000	6038.000	6049.000	6050.000	6051.000	6062.000	6063.000	6066.000
6070.000	6071.000	6072.000	6074.000	6075.000	6078.000	6090.000	6095.000	6096.000	6098.000
6103.000	6104.000	6106.000	6116.000	6136.000	6143.000	6149.000	6151.000	6160.000	6164.000
6167.000	6168.000	6170.000	6172.000	6173.000	6176.000	6179.000	6180.000	6183.000	6187.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                  6188.000 6193.000 6196.000 6197.000 6199.000 6200.000 6201.000 6206.000 6220.000 6221.000
                                  6223.000 6225.000 6228.000 6232.000 6236.000 6240.000 6242.000 6243.000 6244.000 6245.000
                                  6251.000 6252.000 6253.000 6258.000 6262.000 6267.000 6272.000 6273.000 6275.000 6276.000
                                  6286.000 6290.000 6294.000 6296.000 6306.000 6313.000 6321.000 6323.000 6326.000 6333.000
                                  6334.000 6335.000 6351.000 6353.000 6354.000 6355.000 6359.000 6360.000 6369.000 6372.000
                                  6383.000 6384.000 6385.000 6386.000 6394.000 6405.000 6417.000 6432.000 6442.000 6474.000
                                  6498.000 6514.000 6515.000 6533.000 6535.000 6541.000 6550.000 6565.000 6568.000 6588.000
                                  6643.000 6665.000 6670.000 6676.000 6684.000 6695.000 7201.000 7207.000 7209.000 7210.000
                                  7244.000 7275.000 7317.000 7341.000 7344.000 7352.000 7390.000 7410.000 7412.000 7428.000
                                  7440.000 7452.000 7453.000 7454.000 7462.000 7468.000 7471.000 7483.000 7487.000 7528.000
                                  7533.000 7537.000 7555.000 7559.000 7597.000 7618.000 7619.000 7620.000 7622.000 7627.000
                                  7629.000 7639.000 7640.000 7641.000 7661.000 7663.000 7669.000 7683.000 7690.000 7726.000
                                  7745.000 7750.000 7759.000 7785.000 7805.000 7810.000 7818.000 7821.000 7828.000 7829.000
                                  7833.000 7860.000 7882.000 7883.000 7885.000 7902.000 7913.000 7963.000 7965.000 7967.000
                                  7969.000 7980.000 7982.000 7983.000 7985.000 7987.000 7988.000 7990.000 7992.000 7993.000
                                  8029.000 8033.000 8034.000 8035.000 8037.000 8046.000 8052.000 8056.000 8058.000 8059.000

T03SIG                  STRING    REFERENCED AT :   3645.000 3705.000

TBL1                    VARIABLE  REFERENCED AT :    251.000 3794.000 3806.000 3816.000 3889.000 4193.000 4235.000 4288.000
                                  4289.000 4377.000

TBL2                    VARIABLE  REFERENCED AT :    258.000  961.000 3215.000 3220.000 3316.000 3350.000 3424.000 3796.000
                                  3808.000 3817.000 3973.000 3998.000 4292.000

TBL3                    VARIABLE  REFERENCED AT :    264.000 3797.000 3809.000 3818.000 3981.000 3999.000 4293.000

TBL4                    VARIABLE  REFERENCED AT :    270.000 3985.000 3994.000 4294.000

TBL5                    VARIABLE  REFERENCED AT :    276.000 4295.000

TBL6                    VARIABLE  REFERENCED AT :    282.000 4296.000

TBL7                    VARIABLE  REFERENCED AT :    288.000 3931.000 3932.000 3943.000 3962.000 3965.000 4208.000 4297.000
                                  4301.000

TBL8                    VARIABLE  REFERENCED AT :    292.000 3929.000 3930.000 3942.000 3961.000 3964.000 4210.000 4298.000
                                  4305.000

TBM                     LABEL     REFERENCED AT :   4993.000 4999.000 5002.000 8300.000

TBR                     LABEL     REFERENCED AT :   4539.000 8253.000

TCW                     VARIABLE  REFERENCED AT :    225.000 1937.000 2040.000 2041.000 2042.000 2055.000 2372.000 2465.000
                                  2468.000 2957.000

TCWA.ERROR.EXIT         LABEL     REFERENCED AT :   2450.000

TD                      LABEL     REFERENCED AT :   2725.000 8577.000 8578.000

TD.2000                 LABEL     REFERENCED AT :   2734.000 2736.000

TD.CLASSF.ERR.EXIT      LABEL     REFERENCED AT :   2719.000 2720.000 2722.000

TD.CLASSF.ERROR         LABEL     REFERENCED AT :   2717.000 2729.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
TD.DEV.CLASS0                   LABEL    REFERENCED AT :   2754.000

TD.DEV.CLASS1                   LABEL    REFERENCED AT :   2747.000 2814.000

TD.DEV.CLASS2                   LABEL    REFERENCED AT :   2744.000 2849.000

TD.DEV.CLASSE                   LABEL    REFERENCED AT :   2735.000 2743.000 2748.000 2879.000

TD.NO.IOC                       LABEL    REFERENCED AT :   2788.000

TD.TEST.STATUS                  LABEL    REFERENCED AT :   2731.000 2739.000 2779.000

TD2                             VARIABLE REFERENCED AT :    210.000 2740.000 2767.000 2772.000 2775.000 2825.000 2860.000 2871.000
                                         2886.000 2945.000

TD2.WRITE                       LABEL    REFERENCED AT :   2776.000 2778.000 2890.000 2954.000

TD2.WRITE.CC                    LABEL    REFERENCED AT :   2723.000 2789.000 2797.000 2809.000 2816.000 2827.000 2828.000 2840.000
                                         2842.000 2851.000 2862.000 2863.000 2872.000 2873.000 2900.000 2958.000 2960.000

TD4.DEV                         LABEL    REFERENCED AT :   2796.000 2823.000 2858.000 2925.000

TD4.ECLASS                      LABEL    REFERENCED AT :   2897.000 2904.000 2915.000

TD8.DEV                         LABEL    REFERENCED AT :   2804.000 2833.000 2869.000 2936.000

TD84                            VARIABLE REFERENCED AT :    206.000 2733.000 2763.000 2897.000 2898.000 2915.000 2916.000 2930.000
                                         2939.000 2941.000

TEMP1                           VARIABLE REFERENCED AT :    159.000 1103.000 1113.000 3406.000 3411.000 5233.000 5242.000 5501.000
                                         5506.000 6660.000 6680.000 7245.000 7250.000 7980.000 7984.000 8029.000 8034.000

TEMP3                           VARIABLE REFERENCED AT :    165.000 1104.000 1111.000 3382.000 3390.000 4776.000 4781.000 4908.000
                                         4911.000 5025.000 5027.000 6670.000 6677.000 8046.000 8061.000

TEMP4                           VARIABLE REFERENCED AT :    168.000  674.000 1114.000 6701.000

TEST.75                         LABEL    REFERENCED AT :   8009.000

TEST.75.INT                     LABEL    REFERENCED AT :    846.000  855.000

TEST.BIBUSY                     LABEL    REFERENCED AT :   1801.000 1974.000 2427.000

TEST.CLASSF.IPL.INT             LABEL    REFERENCED AT :   7712.000 7762.000

TEST.COUNT.1                    LABEL    REFERENCED AT :   7493.000

TEST.COUNT.2                    LABEL    REFERENCED AT :   7506.000

TEST.COUNT.3                    LABEL    REFERENCED AT :   7516.000

TEST.COUNT1.1                   LABEL    REFERENCED AT :   7564.000 7566.000

TEST.COUNT2                     LABEL    REFERENCED AT :   7579.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

```
TEST.COUNT3                    LABEL    REFERENCED AT :   7591.000

TEST.ELSA                      LABEL    REFERENCED AT :   8015.000

TEST.FFINT                     LABEL    REFERENCED AT :    672.000  701.000  722.000  751.000  784.000 1076.000

TEST.FULL.SCALE.NEGATIVE       LABEL    REFERENCED AT :   2964.000 3810.000

TEST.IOCD2                     LABEL    REFERENCED AT :   7698.000 7704.000

TEST.IPL                       LABEL    REFERENCED AT :   2023.000 2026.000

TEST.IPU.CPU                   LABEL    REFERENCED AT :   3157.000 7863.000

TEST.MAPPED.INTERRUPT          LABEL    REFERENCED AT :    957.000  968.000

TEST.MPYDIV.OFUF               LABEL    REFERENCED AT :   3535.000 3539.000

TEST.NON.INTERRUPTABLE         LABEL    REFERENCED AT :    651.000  680.000

TEST.RAM.LOAD                  LABEL    REFERENCED AT :   1309.000 2254.000 2422.000

TEST.RETRY                     LABEL    REFERENCED AT :    827.000  848.000 2022.000 2029.000 2129.000 2138.000 2149.000 2208.000
                                        2219.000 6400.000 6419.000

TEST.TRAP.CALM                 LABEL    REFERENCED AT :   5248.000

TI                             VARIABLE REFERENCED AT :    246.000 1931.000 2397.000 2476.000

TIM.FLG                        VARIABLE REFERENCED AT :    541.000 3110.000 3117.000

TIMER.CONTROL                  LABEL    REFERENCED AT :   1818.000 1820.000

TIMER.EXIT                     LABEL    REFERENCED AT :   1817.000 1819.000 1822.000

TIO                            LABEL    REFERENCED AT :   7296.000

TMAPR                          LABEL    REFERENCED AT :   6652.000 8431.000

TMP0                           VARIABLE REFERENCED AT :    201.000 1305.000 1305.000 1431.000 1459.000 1459.000 1517.000 1520.000
                                        1549.000 1549.000 1593.000 1648.000 1648.000 1664.000 2461.000 2466.000 3595.000 3599.000
                                        3611.000 3633.000 3651.000 3802.000 3836.000 3915.000 3936.000 4026.000 4066.000 4075.000
                                        4207.000 4226.000 4255.000

TMP1                           VARIABLE REFERENCED AT :    205.000 3846.000 3857.000 3917.000 3937.000 4024.000 4040.000 4045.000
                                        4077.000 4192.000 4234.000

TMP2                           VARIABLE REFERENCED AT :    209.000 3660.000 3910.000 3919.000 3921.000 4197.000 4251.000 4260.000

TMP3                           VARIABLE REFERENCED AT :    224.000 1436.000 1461.000 3938.000 3966.000 3968.000 4027.000 4044.000
                                        4064.000 4068.000 4205.000 4209.000 4215.000 4233.000

TMP4                           VARIABLE REFERENCED AT :    229.000 2470.000 2471.000 2968.000 3763.000 3768.000 3813.000 3831.000
                                        3847.000 3850.000 3854.000 3877.000 3891.000 3894.000 3916.000 3922.000 3934.000 3997.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                4025.000 4041.000 4046.000 4078.000 4191.000 4201.000 4206.000 4264.000

TMP5                    VARIABLE REFERENCED AT :    242.000 1382.000 1389.000 1405.000 1425.000 4195.000 4214.000 4222.000
                                4231.000 4231.000 4246.000 4248.000 4258.000

TMP6                    VARIABLE REFERENCED AT :    245.000 1380.000 1384.000 1387.000 1391.000 1396.000 1396.000 1399.000
                                1406.000 1426.000 4219.000 4229.000 4229.000 4244.000

TMP7                    VARIABLE REFERENCED AT :    249.000 1390.000 1427.000

TNIBL                   STRING   REFERENCED AT :   1434.000 1435.000 1457.000 2469.000 2668.000 3594.000 3611.000 3834.000
                                3835.000 3857.000 4097.000 4107.000 4111.000 4118.000 4131.000 4145.000 4148.000 4149.000
                                4150.000 4151.000 4152.000 4276.000 4378.000 4469.000 4611.000 4617.000 4618.000 5758.000
                                5857.000 5957.000 6142.000 6680.000 6682.000 7207.000 7208.000

TNIBR                   STRING   REFERENCED AT :   2917.000 2965.000 2966.000 3617.000 3618.000 3619.000 3620.000 3621.000
                                3622.000 3675.000 3685.000 3811.000 3823.000 3900.000 3914.000 3927.000 4013.000 4042.000
                                4067.000 4122.000 4133.000 4263.000 4277.000 4365.000 4367.000 4368.000 4369.000 4370.000
                                4371.000 4372.000 4373.000 4374.000 4375.000 4377.000 5352.000

TOGRHF                  STRING   REFERENCED AT :   1759.000 2169.000 3731.000 4420.000 4464.000 4465.000 4511.000 4512.000
                                4534.000 4535.000 4560.000 4569.000 4584.000 4595.000 4761.000 5243.000 5271.000 5442.000
                                7908.000

TPR                     LABEL    REFERENCED AT :   5035.000 8551.000

TRACE                   STRING   REFERENCED AT :    675.000 1012.000

TRACE                   VARIABLE REFERENCED AT :    176.000  652.000  652.000  672.000  676.000  677.000  678.000  694.000
                                695.000  698.000  699.000  719.000  719.000  744.000  744.000  767.000  767.000  770.000
                                770.000  775.000  775.000  816.000  865.000  897.000  919.000  919.000  932.000  947.000
                                955.000  970.000 1004.000 1005.000 1005.000 1015.000 1027.000 1028.000 1045.000 1050.000
                               1050.000 1059.000 1062.000 1062.000 1064.000 1072.000 1074.000 1086.000 1125.000 1137.000
                               1137.000 1232.000 1290.000 1290.000 1610.000 1676.000 1682.000 1682.000 1752.000 1755.000
                               1755.000 1762.000 1762.000 2027.000 2100.000 2629.000 2629.000 2634.000 2634.000 3097.000
                               3097.000 3124.000 3124.000 3146.000 3146.000 3173.000 3173.000 3209.000 3209.000 3214.000
                               3214.000 3273.000 3273.000 3422.000 3423.000 5165.000 5275.000 5277.000 5280.000 5416.000
                               5416.000 5429.000 5429.000 5439.000 5439.000 6091.000 6097.000 6098.000 6101.000 6342.000
                               6342.000 6424.000 7881.000 7883.000 8041.000 8041.000

TRACE.INCOMPLETE        LABEL    REFERENCED AT :   1047.000 1061.000

TRACE.RUN               LABEL    REFERENCED AT :    676.000 1043.000

TRACE.STEP              LABEL    REFERENCED AT :   1049.000

TRACEFF                 STRING   REFERENCED AT :    985.000 1030.000 1058.000 1063.000 1536.000 5279.000 5592.000 5595.000

TRAP                    LABEL    REFERENCED AT :    818.000  925.000

TRAP.INT.EXIT           LABEL    REFERENCED AT :    907.000  923.000

TRAP.MODE55             LABEL    REFERENCED AT :   3158.000 7867.000

TRAP.MODE75             LABEL    REFERENCED AT :    980.000  986.000 1851.000 1867.000 3181.000 3208.000 7861.000 7868.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

7914.000

TRAP.TRL	LABEL	REFERENCED AT : 3065.000 3093.000
TRIPLE.DUD	LABEL	REFERENCED AT : 1984.000 2005.000 2006.000 5847.000
TRN	LABEL	REFERENCED AT : 4515.000 8425.000
TRNM	LABEL	REFERENCED AT : 4530.000 8433.000
TRP	LABEL	REFERENCED AT : 5021.000 8550.000
TRSC	LABEL	REFERENCED AT : 5401.000 8435.000
TRSW	LABEL	REFERENCED AT : 5077.000 8102.000 8180.000
TRUE	STRING	REFERENCED AT : 5280.000
TSCR	LABEL	REFERENCED AT : 5410.000 8436.000
TSFILL	STRING	REFERENCED AT : 3617.000 3618.000 3619.000 3620.000 3621.000 3622.000 3675.000 3685.000
TURN.OFF.STOP	LABEL	REFERENCED AT : 6230.000 6238.000 6247.000 6260.000
TY.TD4	LABEL	REFERENCED AT : 2857.000
TY.TD8	LABEL	REFERENCED AT : 2850.000 2868.000
UART.INPUT	LABEL	REFERENCED AT : 5735.000 5737.000
UART.XMIT	LABEL	REFERENCED AT : 5851.000 5854.000 5949.000
UARTDAV	STRING	REFERENCED AT : 749.000 5735.000 5748.000 5948.000 6215.000 6302.000 6363.000
UARTDS	STRING	REFERENCED AT : 5857.000 5859.000
UARTERR	STRING	REFERENCED AT : 5738.000
UARTTBMT	STRING	REFERENCED AT : 749.000 5851.000 5946.000
UEI	LABEL	REFERENCED AT : 4387.000 8382.000
UEI.REI.EXIT	LABEL	REFERENCED AT : 4390.000 4400.000
UF.ROUTE	LABEL	REFERENCED AT : 3548.000 3553.000
UNBLOCK	STRING	REFERENCED AT : 801.000 965.000 966.000 970.000 1128.000 1236.000 1608.000 3139.000 3140.000 3218.000 3425.000 4389.000 4398.000 4399.000 5346.000 5347.000 5370.000 5371.000 5372.000 5432.000 6384.000 6385.000 6426.000 7452.000 7458.000 7727.000 7962.000 7991.000 8020.000 8042.000 8043.000
UNBLOCKED.75.RETURN	LABEL	REFERENCED AT : 903.000 970.000
UNDEF	LABEL	REFERENCED AT : 4443.000 5269.000 5272.000 5555.000 5602.000 5606.000 8244.000 8245.000 8246.000 8247.000 8248.000 8249.000 8254.000 8255.000 8256.000 8258.000 8259.000 8269.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
                                8270.000 8271.000 8272.000 8273.000 8274.000 8275.000 8276.000 8277.000 8278.000 8279.000
                                8280.000 8281.000 8282.000 8291.000 8301.000 8316.000 8322.000 8323.000 8325.000 8326.000
                                8327.000 8328.000 8329.000 8354.000 8356.000 8358.000 8359.000 8362.000 8363.000 8364.000
                                8365.000 8367.000 8406.000 8407.000 8408.000 8409.000 8410.000 8411.000 8412.000 8413.000
                                8422.000 8423.000 8427.000 8446.000 8447.000 8448.000 8449.000 8450.000 8451.000 8454.000
                                8455.000 8456.000 8457.000 8458.000 8459.000 8469.000 8470.000 8471.000 8472.000 8473.000
                                8474.000 8477.000 8478.000 8479.000 8480.000 8481.000 8482.000 8490.000 8498.000 8499.000
                                8502.000 8503.000 8522.000 8523.000 8524.000 8525.000 8526.000 8527.000 8528.000 8529.000
                                8552.000 8553.000 8554.000 8555.000 8556.000 8557.000 8558.000 8559.000 8592.000 8593.000
                                8594.000 8600.000 8601.000 8602.000 8616.000 8617.000 8618.000 8620.000 8621.000 8622.000
                                8624.000 8625.000 8626.000 8628.000 8629.000 8630.000 8639.000 8641.000 8643.000 8645.000
                                8647.000 8649.000 8651.000 8653.000 8662.000 8664.000 8666.000 8668.000 8670.000 8672.000
                                8674.000 8676.000

UNDEF.55                        LABEL    REFERENCED AT :   3409.000 4409.000 4422.000 4428.000 7202.000

UNDEF.55.5XX                    LABEL    REFERENCED AT :   3408.000 3450.000 3487.000 3497.000 7897.000 7899.000

UNDEF.55.FLG                    VARIABLE REFERENCED AT :    549.000 4429.000

UNDEF.75                        LABEL    REFERENCED AT :   4412.000 4421.000 5273.000 6450.000 6740.000 6867.000 7204.000 7238.000
                                         7288.000 7292.000 7319.000

UNDEF.75.1                      LABEL    REFERENCED AT :   4419.000

UNDEF.75.FLG                    VARIABLE REFERENCED AT :    540.000 2435.000 4422.000 8021.000

UNDEF.XX                        LABEL    REFERENCED AT :   4431.000 4440.000

UNDEF1                          LABEL    REFERENCED AT :   5171.000 5187.000 5267.000 8185.000 8190.000 8191.000 8192.000 8193.000
                                         8194.000 8195.000 8204.000 8205.000 8206.000 8207.000 8208.000 8209.000 8210.000 8211.000
                                         8212.000 8213.000 8214.000 8215.000 8216.000 8217.000 8218.000 8219.000 8220.000 8221.000
                                         8222.000 8223.000 8224.000 8225.000 8226.000 8227.000 8228.000 8229.000 8230.000 8231.000
                                         8232.000 8233.000 8234.000 8235.000

UNDEF2                          LABEL    REFERENCED AT :   5270.000 8112.000 8113.000 8114.000 8115.000 8116.000 8117.000

UNDEFINED.INSTRUCTION           LABEL    REFERENCED AT :   1073.000 3145.000

UNDERFLOW                       LABEL    REFERENCED AT :   3555.000 3562.000

UNEXP.M.ERR.FLG                 VARIABLE REFERENCED AT :    547.000 3081.000

UNEXPLAINED                     LABEL    REFERENCED AT :   3059.000 3061.000

UNEXPLAINED.ERR                 LABEL    REFERENCED AT :   3063.000 3080.000

UPACK                           STRING   REFERENCED AT :   1020.000 1035.000 1059.000 1272.000 1534.000 1732.000

UPDATE.ACTIVE                   LABEL    REFERENCED AT :   2231.000 2242.000

UPDATE.CPU.STATUS               LABEL    REFERENCED AT :   1864.000 3133.000 3153.000 3178.000 3204.000 3240.000 3347.000 4430.000
                                         7857.000 7880.000 7911.000

UPDATE.INT.TBL.EXIT             LABEL    REFERENCED AT :   7317.000 7350.000 7659.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

```
UPDATE.INTERRUPT          LABEL    REFERENCED AT :  2153.000 2225.000 2230.000 2239.000 2240.000 2245.000 2246.000
UPREQ                     STRING   REFERENCED AT :   746.000  772.000
USE.CURRENT.PSW           LABEL    REFERENCED AT :  6383.000 6423.000
USE.CURRENT.TEST.MAP      LABEL    REFERENCED AT :  5354.000 5357.000
USECAR                    STRING   REFERENCED AT :  3574.000 3767.000 3837.000 3906.000 3921.000 3924.000 3931.000 3943.000
                                   3962.000 3965.000 4003.000 4008.000 4010.000 4078.000 4085.000 4231.000 4246.000 4260.000
                                   4273.000 4306.000 4657.000 4687.000 4727.000 4941.000
V                         STRING   REFERENCED AT :  4576.000
WAIT                      LABEL    REFERENCED AT :  5424.000 5434.000 8376.000
WAIT.BLK.MODE.ERR         LABEL    REFERENCED AT :  5415.000 5433.000
WAIT.EXIT                 LABEL    REFERENCED AT :  5431.000 5436.000
WAIT.EXIT1                LABEL    REFERENCED AT :  5437.000 5441.000
WAIT.INTR                 LABEL    REFERENCED AT :  5438.000
WAIT.LOOP                 LABEL    REFERENCED AT :  5430.000 5435.000
WAIT.RDY                  LABEL    REFERENCED AT :  2131.000
WAIT.RTN                  LABEL    REFERENCED AT :  5427.000 6808.000
WCS(S)                    STRING   REFERENCED AT :  5502.000 5507.000 5531.000 5533.000 5617.000 7785.000 7800.000
WCS.ADDR                  VARIABLE REFERENCED AT :  5485.000 5490.000
WCS.ADDR.OUT.OF.RANGE     LABEL    REFERENCED AT :  5549.000
WCS.ERROR                 LABEL    REFERENCED AT :  5505.000 5511.000 5531.000
WCS.EXIT                  LABEL    REFERENCED AT :  5510.000 5534.000 5551.000
WCS.NOT.PRESENT           LABEL    REFERENCED AT :  5543.000 5552.000 5580.000
WCS.NOT.PRESENT1          LABEL    REFERENCED AT :  5566.000 5579.000
WCS.RETURN.EXIT           LABEL    REFERENCED AT :  5576.000
WCS.SETUP                 LABEL    REFERENCED AT :  5499.000 5521.000 5539.000
WCS.SYS.RESET             LABEL    REFERENCED AT :  5594.000 5596.000
WCWCS                     LABEL    REFERENCED AT :  7320.000
WCWCS1                    LABEL    REFERENCED AT :  7322.000 7429.000
```

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

WCWCS2 LABEL REFERENCED AT : 7430.000 7436.000

WDOT STRING REFERENCED AT : 1991.000 2425.000 7408.000 7652.000

WRITE STRING REFERENCED AT : 885.000 951.000 963.000 1110.000 1964.000 2369.000 2394.000 2397.000
2477.000 2959.000 3309.000 3315.000 3319.000 3322.000 3326.000 4843.000 4847.000 4870.000
4877.000 4895.000 4912.000 4940.000 4945.000 4950.000 4952.000 4975.000 4981.000 5224.000
5503.000 5509.000 5892.000 7530.000 7600.000 7839.000

WRITE.WCS LABEL REFERENCED AT : 7783.000

WRPMAP STRING REFERENCED AT : 1184.000 5027.000

WRT.RD.PSW.EXIT LABEL REFERENCED AT : 1685.000 1689.000

WWCS LABEL REFERENCED AT : 5520.000 8387.000

X STRING REFERENCED AT : 2522.000 3450.000 3460.000 3472.000 3474.000 3533.000 4409.000 4630.000
4869.000 4962.000 4991.000 5013.000 5061.000 5089.000 5162.000 5186.000 6719.000 7169.000
8102.000 8106.000 8112.000 8113.000 8114.000 8115.000 8116.000 8117.000 8118.000 8119.000
8126.000 8136.000 8180.000 8184.000 8185.000 8190.000 8191.000 8192.000 8193.000 8194.000
8195.000 8196.000 8197.000 8244.000 8245.000 8246.000 8247.000 8248.000 8249.000 8254.000
8255.000 8256.000 8257.000 8258.000 8259.000 8291.000 8301.000 8314.000 8315.000 8316.000
8318.000 8322.000 8323.000 8355.000 8357.000 8366.000 8375.000 8376.000 8377.000 8379.000
8380.000 8381.000 8382.000 8383.000 8385.000 8388.000 8389.000 8390.000 8401.000 8402.000
8403.000 8404.000 8405.000 8406.000 8407.000 8408.000 8409.000 8410.000 8411.000 8412.000
8413.000 8422.000 8423.000 8427.000 8428.000 8430.000 8431.000 8435.000 8436.000 8469.000
8470.000 8471.000 8472.000 8473.000 8474.000 8477.000 8478.000 8479.000 8480.000 8481.000
8482.000 8490.000 8498.000 8499.000 8502.000 8503.000 8543.000 8592.000 8593.000 8594.000
8600.000 8601.000 8602.000

XCR LABEL REFERENCED AT : 4492.000 8426.000

XCRM LABEL REFERENCED AT : 4497.000 8434.000

XCRM2 LABEL REFERENCED AT : 4493.000 4500.000

XMIT.EXIT LABEL REFERENCED AT : 5852.000

YY.Y LABEL REFERENCED AT : 7492.000 7499.000

ZBM LABEL REFERENCED AT : 4974.000 8298.000

ZE STRING REFERENCED AT : 821.000 822.000 835.000 846.000 871.000 934.000 946.000 1024.000
1087.000 1101.000 1121.000 1128.000 1220.000 1222.000 1268.000 1309.000 1327.000 1520.000
1702.000 1724.000 1772.000 1812.000 1844.000 1899.000 1911.000 1928.000 1940.000 2042.000
2067.000 2077.000 2092.000 2124.000 2127.000 2142.000 2144.000 2152.000 2153.000 2183.000
2201.000 2206.000 2217.000 2235.000 2257.000 2394.000 2422.000 2461.000 2470.000 2499.000
2516.000 2548.000 2663.000 2677.000 2682.000 2710.000 2789.000 2890.000 2971.000 2981.000
3211.000 3229.000 3357.000 3421.000 3476.000 3522.000 3743.000 3991.000 3999.000 4473.000
4637.000 4789.000 4818.000 4827.000 4835.000 4849.000 4871.000 4880.000 4881.000 4897.000
4922.000 4944.000 4969.000 4998.000 5040.000 5194.000 5211.000 5255.000 5311.000 5353.000
5455.000 5510.000 5911.000 6394.000 6405.000 6417.000 6498.000 6541.000 6568.000 6588.000
6695.000 7317.000 7440.000 7487.000 7533.000 7559.000 7629.000 7640.000 7810.000 8052.000
8058.000

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

ZERO	VARIABLE	REFERENCED AT : 299.000 1233.000 2377.000 3635.000 3641.000 3683.000 3690.000 3694.000 3699.000 3702.000 3832.000 3905.000 3906.000 4002.000 4003.000 4055.000 4073.000 4084.000 4085.000 4091.000 4147.000 4272.000 4273.000 4306.000
ZM	LABEL	REFERENCED AT : 4842.000 8543.000
ZM.INDIRECT	LABEL	REFERENCED AT : 4845.000 4851.000

* * * * *

```
$ASSIGN1 SI=DEF.75H
$OPTION 2
$ALLOCATF 20000
$EXECUTE MICRO
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
                                                                                                                    0001.000
                                    *** DEF.75H ***                                                                 0002.000
                                    *** 08 NOVEMBER 1979****                                                        0003.000
                                    *** WHSDEF24 64 BIT DEFINITIONS FOR THE 32/75 CPU WITH S/A OR IPU ***           0004.000
                                    @       $FR=0  $XT=@1F $XC=0 ;                                                  0005.000
                                            $SIZE   64                                                              0006.000
0001000000000000                    .$DEV   $EQ     @0001000000000000  DEFAULT IMAGE                                0007.000
                                                                                                                    0008.000
                                    *** MICROWORD FIELD DEFINITIONS ***                                             0009.000
FFFFFFFFFFFFFFFF                    NOP.F   $FD     0,64                                                            0010.000
00000F0000000000                    D       $FD     40,4            DEST                                            0011.000
00000FE000000000                    DF      $FD     40,4;37,3       DEST & FREAD                                    0012.000
000000E00F000000                    RF      $FD     24,4;37,3       REGNO & FREAD                                   0013.000
00000000F0000000                    XO      $FD     28,4            X, U, S ORDERS AND W, S TESTS                   0014.000
0000001F00000000                    YO      $FD     32,5            Y ORDER                                         0015.000
00000000000F0000                    CONDO   $FD     16,4            CONDITIONAL ORDER                               0016.000
003800000F00000                     SHIFT   $FD     20,4;51,3       X ORDER,AMUX                                    0017.000
000000  00F00 00                    S C     $FD     20,4            SHIFT CODE                                      0018.000
0000c0  0F  0   0                   C       $FD     24,4            COND CODE                                       0019.000
00000F  0F  0   0                   R       $FD     24,4;40,        REGNO & DEST                                    0020.000
00 00   00  0   00                  A       $FD     51,5            AMUX                                            0021.000
000000  00  0   00                  F R     $FD     37,5            FREAD                                           0022.000
700000  00  0   00                  F ER    $FD     37,3;60,        FREAD AND TEST                                  0023.000
003800  00  0   00                  A YO    $FD     32,5;51,        AMUX & YORDER                                   0024.000
000000  F0  0   00                  Y XO    $FD     28,4;32,        Y-ORDER & X-ORDER                               0025.000
003800  0FF 0   00                  LIT     $FD     20,8;51,3       LITERAL & AMUX                                  0026.000
003800  F0FF 0   00                 LLIT    $FD     20,8;51,3;32,5  LONG LITERAL, AMUX, & Y-ORDER                   0027.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
0038001F0FF00000                BLIT    $FD     20,8;51,3;32,5  BYTE LITERAL, AMUX, & Y-ORDER           0028.000
0007000000000000                B       $FD     48,3            BMUX                                    0029.000
0000F00000000000                ALUF    $FD     44,4            ALU FIELD                               0030.000
F000000000000000                TESTF   $FD     60,4            NORMAL TEST FIELD                       0031.000
000000000F000000                ZTESTF  $FD     24,4            ZTEST FIELD                             0032.000
0000000000700000                ETB4    $FD     20,3            EXTENDED TEST GROUP 4 (ENABLE=BIT 40)   0033.000
0000000000700000                ETB0    $FD     20,3            EXTENDED TEST GROUP 0 (ENABLE=BIT 39)   0034.000
0000000000700000                ETB1    $FD     20,3            EXTENDED TEST GROUP 1 (ENABLE=BIT 38)   0035.000
0000000000700000                ETB2    $FD     20,3            EXTENDED TEST GROUP 2 (ENABLE=BIT 37)   0036.000
0000000000700000                ETB3    $FD     20,3            EXTENDED TEST GROUP 3 (ENABLE=BIT 36)   0037.000
000000000F800000                EXTST   $FD     23,5            ENABLE EXTENDED TEST GROUPS 0-4         0038.000
0E000000000F0000                HOP     $FD     16,4;57,3       HOP                                     0039.000
0E00000000FF0000                LEAP    $FD     16,8;57,3       LEAP                                    0040.000
0F0000000FFF0000                BRANCH  $FD     16,12;57,3      BRANCH                                  0041.000
0E0000001FFF0000                WCS.BR  $FD     16,13;57,3      BRANCH 13 IN OR OUT OF WCS              0042.000
0E0000000000000                 SC      $FD     57,3            SEQUENCE CONTROL                        0043.000
01C0000000F00000                PG      $FD     20,4;54,3       DECODE PAGE                             0044.000
0000000007000000                FF1     $FD     24,3            FLIP FLOP NUMBER (GROUP 1)              0045.000
0000000008000000                S.R.FF1 $FD     27,1            SET/RESET FLIP FLOP (GROUP 1)           0046.000
0000000007000000                FF2     $FD     24,3            FLIP FLOP NUMBER (GROUP 2)              0047.000
0000000008000000                S.R.FF2 $FD     27,1            SET/RESET FLIP FLOP (GROUP 2)           0048.000
0000000007000000                FF3     $FD     24,3            FLIP FLOP NUMBER (GROUP 3)              0049.000
0000000008000000                S.R.FF3 $FD     27,1            SET/RESET FLIP FLOP (GROUP 3)           0050.000
01C0000000000000                CONTROL $FD     54,3            CONTROL FIELD                           0051.000
                                                                                                        0052.000
                                *** FLOATING POINT FIELD DEFINITIONS                                    0053.000
000000000000C000                FADDR   $FD     14,2    FRACTION ADDER CONTROL FIELD                    0054.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SFL 32/75 CPU

```
PC   MICROWORD

     0000000000000000          FAMUX   $FD     14,0    FRACTION AMUX SELECT FIELD (DUMMY FIELD)          0055.000
     0000000000003000          FBMUX   $FD     12,2    FRACTION BMUX SELECT FIELD                        0056.000
     0000000000003000          FSIB    $FD     12,2    SHIFTER INPUT BUS SELECT FIELD                    0057.000
     0000000000000F00          FDEST   $FD     9,3     FRACTION DESTINATION SELECT FIELD                 0058.000
     0000000000000800          SPDEST  $FD     11,1    SPECIAL SHIFT FRACTION DESTINATION SELECT         0059.000
     0000000000000600          SPMPY   $FD     9,2     SPECIAL SHIFT MULTIPLY DESTINATION OVERRIDE       0060.000
     00000000000001C0          FDMUX   $FD     6,3     FRACTION DMUX SELECT FIELD                        0061.000
     0000000000000020          EA      $FD     5,1     EXPONENT ADDER CONTROL FIELD                      0062.000
     0000000000000010          BEAI    $FD     4,1     B EXPONENT ADDER INPUT MUX SELECT FIELD           0063.000
     000000000000000C          AEAI    $FD     2,2     A EXPONENT ADDER INPUT MUX SELECT FIELD           0064.000
     0000000000000003          EDEST   $FD     0,2     EXPONENT DESTINATION SELECT FIELD                 0065.000
                                                                                                         0066.000
                               *** BIT 00-47 FIELD REDEFINITIONS ***                                     0067.000
     8000000000000000          FPGRP1  $FD     63,1            ENABLE FP GROUP 1 ORDERS                  0068.000
     D000000000000000          FPGRP2  $FD     60,1;62,2       ENABLE FP GROUP 1 &2 ORDERS               0069.000
     F000000000000000          FPVJMP  $FD     60,4            ENABLE FP VECTORED JUMP                   0070.000
     0030000000000000          FCNTSEL $FD     52,2                                                      0071.000
     000C000000000000          FSHFTYP $FD     50,2                                                      0072.000
     0003F00000000000          FPCNT   $FD     44,6                                                      0073.000
     000000000F000000          FPORDF2 $FD     24,4            ORDER GROUP 2                             0074.000
     0000000000F00000          FPORDF1 $FD     16,4            ORDER GROUP 1                             0075.000
                                                                                                         0076.000
                               *** PSEUDO FIELD DEFINITIONS ***                                          0077.000
                               AR      $PD     [RF]4;3,0       AMUX REG                                  0078.000
                               BR      $PD     [RF]4;3,0       BMUX REG                                  0079.000
                               RN      $PD     [R]4;5,=8       DEST REG                                  0080.000
                               FR      $PD     [FFR]3;1,1;3,=6;1,0                                       0081.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
LIT(@17) $PD    [LIT]8;24,@FFFFFF      SHORT LITERAL                    0082.000
LIT(@14) $PD    [LLIT]24,@FFFFFF;8                                      0083.000
LIT(@15) $PD    [LLIT]16,@FFFF;8;8,@FF                                  0084.000
LIT(@16) $PD    [LLIT]8,@FF;8;16,@FFFF                                  0085.000
LIT(@13) $PD    [LLIT]8;24,0                                            0086.000
LIT(@10) $PD    [LLIT]24,0;8                                            0087.000
LIT(@11) $PD    [LLIT]16,0;8;8,0                                        0088.000
LIT(@12) $PD    [LLIT]8,0;8;16,0                                        0089.000
BLIT(@11) $PD   [BLIT]8          SCRATCH PAD AMUX LITERAL               0090.000
PG(4)  $PD      [PG]4;1,0        LOWER DECODE ORDER                     0091.000
PG(5)  $PD      [PG]4;2,1        UPPER DECODE ORDER                     0092.000
HOP(4) $PD      [HOP]4;12,($+1)↑-4                                      0093.000
LEAP(5) $PD     [LEAP]8;8,($+1)↑-8                                      0094.000
BRANCH(6) $PD   [BRANCH]12;4,($+1)↑-12                                  0095.000
GOTO(4) $PD     [HOP]4;12,($+1)↑-4                                      0096.000
GOTO(5) $PD     [LEAP]8;8,($+1)↑-8                                      0097.000
GOTO(6) $PD     [BRANCH]12;4,($+1)↑-12                                  0098.000
GOTO(7) $PD     [WCS.BR]13;1,0                                          0099.000
                                                                        0100.000
*** FLOATING POINT PSEUDO FIELD DEFINITIONS ***                         0101.000
FPCNT  $PD      [FPCNT]6                                                0102.000
FPCNT  $PD      [FPCNT]6;1,1                                            0103.000
                                                                        0104.000
*** BINARY OUTPUT FORMAT***                                             0105.000
F1      $FMT    1,60,4;3,57,3;5,54,3;7,51,3;9,48,3;11,44,4;             0106.000
                13,40,4;15,37,3;18,32,5;20,28,4;24,16,12;26,14,2;       0107.000
                28,12,2;30,9,3;32,6,3;34,5,1;36,4,1;38,2,2;40,0,2;      0108.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
                45,[(GOTO)];45,[(HOP)];45,[(LEAP)];45,[(BRANCH)];          0109.000
                45,[(WCS.BR)]                                              0110.000
                                                                           0111.000
*** AMUX REGISTER DEFINITIONS ***                                          0112.000
S       $RD     [A]0                                                       0113.000
SLEFT   $RD     [A]1                                                       0114.000
SNIBL   $RD     [A]2                                                       0115.000
SNIBR   $RD     [A]3                                                       0116.000
STATUS  $RD     [A]5                                                       0117.000
BMG     $RD     [A]6                                                       0118.000
                                                                           0119.000
*** BMUX REGISTER DEFINITIONS ***                                          0120.000
T       $RD     [B]0                                                       0121.000
MAR     $RD     [B]1                                                       0122.000
N       $RD     [B]1                                                       0123.000
IO      $RD     [B]2                                                       0124.000
DI      $RD     [B]3                                                       0125.000
INTLVL  $RD     [B]6                                                       0126.000
PNLDATA $RD     [B]6                                                       0127.000
                                                                           0128.000
*** FLOATING POINT FRACTION AMUX REGISTER DEFINITION ***                   0129.000
RA      $RD     [FAMUX]0        DUMMY FP AMUX                              0130.000
                                                                           0131.000
*** FLOATING POINT FRACTION BMUX REGISTER DEFINITIONS ***                  0132.000
RB      $RD     [FBMUX]0                                                   0133.000
RA      $RD     [FBMUX]1                                                   0134.000
RP      $RD     [FBMUX]2                                                   0135.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU

PC MICROWORD

```
                                                                                0136.000
*** FLOATING POINT EXPONENT AMUX REGISTER DEFINITIONS ***                       0137.000
RAE     $RD     [AEAI]0                                                         0138.000
RBE     $RD     [AEAI]1                                                         0139.000
EXPK    $RD     [AEAI]2                                                         0140.000
RMEXP   $RD     [AEAI]3                                                         0141.000
                                                                                0142.000
*** FLOATING POINT EXPONENT BMUX REGISTER DEFINITIONS ***                       0143.000
RBE     $RD     [BEAI]0                                                         0144.000
RAE     $RD     [BEAI]1                                                         0145.000
                                                                                0146.000
*** PATTERN DEFINITIONS ***                                                     0147.000
@DREG   ( 'NOD' : 'S' : 'PC' : 'MAR' : 'LOCSTORE' : 'I1' : 'DI' :              0148.000
        'MARIX' )@D#$. :                                                        0149.000
        ( 'NU' : 'NL' : 'FULLMAR' : 'T' )@D#$.+@C ;                             0150.000
@DEST   ( ( ( 'FR('@D#@8 #.(FR)=$.FV:6↑4 $FR=4 :                               0151.000
        'R(' ( 'RD'@DF#@7B : 'RO'@DF#@6B : 'R'@D#9 : 'NCTR'@DF#@3B :            0152.000
        'S'@DF#@5B : 'X'@DF#@2B : 'DIX'@DF#@1B : #.(RN)R=$.FV:@8O ) )           0153.000
        [ ',' ( 'RH' : 'LH' )@YO#$.+13 ] ')' ) : @DREG :                        0154.000
        'WCS(S)'@XO#7 ) " '=' " ;                                               0155.000
@SAMUX  'BMGMR'@AYO#6↑5+4 ;                                                     0156.000
@REGS   'RD'@FR#7 : 'RO'@FR#6 : 'R'@FR#4 : 'NCTR'@FR#3 : 'S'@FR#5 :             0157.000
        'X'@FR#2 : 'DIX'@FR#1 ;                                                 0158.000
@AMUX   'FR('@A#4 #.(FR)=$.FV:6↑4 $FR=4 ')' :                                   0159.000
        'R('@A#4 ( @REGS : #.(AR)RF ) ')' : @SAMUX : #.(LIT)A ;                 0160.000
@BMOD   'ZE'@XO#@F : 'SF'@XO#@E  :                                              0161.000
@HWS    'HWS' =5 ;                                                              0162.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
@BMUXR  ( 'FR(' #.(FR)=$.FV:6↑4 $FR=4 :                                  0163.000
        'R(' ( @REGS : #.(BR)RF ) )                                       0164.000
        =4 [ ',' ( @BMOD : @HWS ) [ ',' ( @HWS : @BMOD ) ]                0165.000
        ]@B#$.$ ')' ;                                                     0166.000
@BMUX   @BMUXR : B [ '(' @BMOD ')' ] ;                                    0167.000
@BMUX.NAMES @BMUXR : ( 'IO'.B : 'DI'.B : 'T'.B : 'MAR'.B :                0168.000
        'N'.B : 'INTLVL'.B : 'PNLDATA'.B ) [ '(' @BMOD ')' ] ;            0169.000
                                                                          0170.000
*** TEST FIELD PATTERN ***                                               0171.000
@TEST   ( %FAIL : %FAIL : %FAIL : %FAIL : %FAIL : %FAIL : %FAIL :        0172.000
        'NOEXTUNIV' : %FAIL : %FAIL : %FAIL : 'ALUZ' :                    0173.000
        'NALUZ' : %FAIL : %FAIL : 'FALSE' )@TESTF#$. ;                    0174.000
                                                                          0175.000
*** W TEST FIELD PATTERN ***                                             0176.000
@WTEST  ( 'EXTLW' : 'SIGNSAVE' : 'ALUNEGW' : 'FFRUN' :                   0177.000
         'BMUX00' : 'NORC' : %FAIL : 'LFC1V' :                            0178.000
         'BMUX16' : 'BMUX17' : 'BMUX18' : 'BMUX19' :                      0179.000
         'LATERRW' : 'DWORD' : 'HWORD' : 'BYTE' )@XO#$. ;                 0180.000
                                                                          0181.000
*** S TEST FIELD PATTERN ***                                             0182.000
@STEST ( '%BLKTIMEOUT' : '%WCS' : 'IPU' : 'IPU.START' : %FAIL : %FAIL    0183.000
        : %FAIL : %FAIL : 'UNBLOCK' : 'ELSA' : 'ENBL.AEXP' : %FAIL :      0184.000
        'MODE75S' : 'IPU.HALT' : %FAIL : 'EXTMAPERR' )@XO#$. ;            0185.000
                                                                          0186.000
*** Z-TEST FIELD PATTERN ***                                             0187.000
@ZTEST ( 'TRUE' : 'EXTL' : 'NHMPREV' : 'MODE75' :                         0188.000
        'SIGNSAVEZ' : 'CCTEST' : 'LATERRZ' : 'ALU4-7Z' :                  0189.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
                'ALUNEG' : 'NCTR2' : 'RHFLAG' : 'NCTR4' :                          0190.000
                'INDIR' : 'NCTR0' : 'FC1V' : %FAIL )@ZTESTF#$. ;                   0191.000
                                                                                   0192.000
        *** EXTENDED TEST FIELD PATTERNS ***                                       0193.000
        @TEST4  ( 'UARTERR' : 'UARTDAV' : 'FFSYSR' : 'ENRL55' : 'SERIAL.PANEL' :   0194.000
                 'OPRNDPE' : 'MAPINVALID' : 'MAPMODF' )@ETR4#$. $XT=$XT&@1E ;      0195.000
        @TEST3  ( 'HIREG' : 'BAUDSCALE' : 'IORESPRDY' : 'IONORESP' : 'INSTNORESP'  0196.000
                 : 'OPNORESP' : 'PROTV' : 'SONET0' )@ETR3#$. $XT=$XT&@0F ;         0197.000
        @TEST2 ( 'PRIVBIT' : 'PWRFAIL' : 'UPPRED' : 'IOCHBUSY' : 'T03SIG' :        0198.000
                'BIBUSY' : 'LATERR' : 'AEXP' )@ETR2#$. $XT=$XT&@17 ;               0199.000
        @TEST1 ('EXFLAG' : 'UARTTBMT' : 'FFINT' : 'IOTIMEOUT' : 'INSTTIMEOUT'      0200.000
                : 'OPTIMEOUT' : 'FLAG' : 'INTRENA'  )@ETB1#$. $XT=$XT&@1B ;        0201.000
        @TEST0 ( 'TRACE' : 'PPATTN' : 'IPLSW' : 'IORETRY' : 'INSTMIUER' :          0202.000
                'OPMIUER' : 'EXTG' : 'ADDRSTOP' )@ETB0#$. $XT=$XT&@1D ;            0203.000
        @XTEST ( @TEST0 : @TEST1 : @TEST2 : @TEST3 : @TEST4 ) [ ':'                0204.000
                ( @TEST0 : @TEST1 : @TEST2 : @TEST3 : @TEST4 ) [ ':'               0205.000
                ( @TEST0 : @TEST1 : @TEST2 : @TEST3 : @TEST4 ) [ ':'               0206.000
                ( @TEST0 : @TEST1 : @TEST2 : @TEST3 : @TEST4 ) [ ':'               0207.000
                ( @TEST0 : @TEST1 : @TEST3 : @TEST3 : @TEST4 ) ] ] ] ] ;           0208.000
                                                                                   0209.000
        *** FIRMWARE FLIP/FLOP PATTERNS (GROUP 1) ***                              0210.000
        @ALTERFF1 ( 'RESET(' : 'SET(' )@S.P.FF1#$. ( 'HIREG' : ( 'EXFF' :          0211.000
                  'EXFLAG' ) : 'PRIV' : 'TRACEFF' : 'OPEFF' : 'DINTRA' :           0212.000
                  'DWAIT' : 'ENAINTFF' )@FF1#$. ')'@CONDO#4 ;                      0213.000
                                                                                   0214.000
        *** FIRMWARE FLIP/FLOP PATTERNS (GROUP 2) ***                              0215.000
        @ALTERFF2 ( 'RESET(' : 'SET(' )@S.P.FF2#$. ( 'ENAUORD' : 'UARTDS' :        0216.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
SFL 32/75 CPU
PC MICROWORD

```
                'UARTDAV' : 'MAPMODE' : 'ENATBMT' : 'IPU.HALT' : 'IPU.START' : 217.000
                'FLAG' )@FF2#$. ')'@CONDO#5 ;                                  0218.000
                                                                               0219.000
*** FIRMWARE FLIP/FLOP PATTERNS (GROUP 3) ***                                  0220.000
@ALTERFF3 ( 'RESET(' : 'SET(' )@S.P.FF3#$. ( 'MODE75' : 'UNBLOCK' :            0221.000
        'ELSA' : 'ENBL.AEXP' : %FAIL : 'ENASORD' :                             0222.000
        'FPDWORD' :  'DIS.BLK.TIMEOUT' )@FF3#$.                                0223.000
        ')'@CONDO#6 ;                                                          0224.000
                                                                               0225.000
*** CONDITIONAL ORDER PATTERN ***                                              0226.000
@CO     ( %FAIL : %FAIL : 'FORCFZ' : ( 'CLRSYSR' : 'IGNSTOP' ) :               0227.000
        %FAIL : %FAIL : %FAIL : 'CLRS' :                                       0228.000
       'ABSDI' : 'ABST' : 'DIVIDE' : 'MPY' : %FAIL : 'CLDNU' :                 0229.000
       'CDECRN' : 'SETAEXP' )@CONDO#$. : @ALTERFF1 : @ALTERFF2 :               0230.000
       @ALTERFF3 : 'SETCC(#)'@CONTROL#1 : 'SETCC(' ( 'AL' : 'V' :              0231.000
       'F' : 'D' : 'BIT' : 'S' : 'AEXP' : 'C' : 'FPU' )@CC#$.                  0232.000
       ')'@CONDO#1 : 'SETXCC(' ( 'S' )@CC#5 ')'@CONDO#@C ;                     0233.000
                                                                               0234.000
*** SEQUENCE CONTROL PATTERNS ***                                              0235.000
*** EXT CONTROL AND ZTEST PATTERN***                                           0236.000
[,F1]  ( [ 'IF' " [ '%' $XC=@8 ] @ZTEST  [ " @CO ] , ] ( 'REGSEL' :            0237.000
       '*JUMPS' : 'MPROM' : 'DIVMSW' : 'DIVLSW' : 'REPEAT' :                   0238.000
       'SCALE' : 'NORM' )@CONTROL#$. $XC=$XC:@6 )@TESTF#$XC ; ;                0239.000
                                                                               0240.000
[,F1]  [ 'IF' " ( @TEST : ( '%' : %MATCH)@TESTF#10-$.*8+$FR @ZTEST             0241.000
       [ " @CO ] :  ( '%' : %MATCH)@TESTF#9-$.*8 @XTEST@EXTST#$XT              0242.000
       [ " @CO ] : ( '%' : %MATCH)@TESTF#5-$.*2 @WTEST [ " @CO ] :             0243.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU

PC MICROWORD

```
                @STEST@TESTF#4 ( " @CO ] ) ] ( " '*' ( 'JUMPD'@SC#2 :                0244.000
                'HOP' " #.(HOP) : 'LEAP' " #.(LEAP) : 'BRANCH' " #.(BRANCH) :        0245.000
                'JUMPJ'@SC#1 : 'JUMPZ'@SC#3 : ( 'GOTO' : 'GO TO' ) " #.(GOTO) :      0246.000
                'LINK'@CONTROL#7 " #.(GOTO) ) ] , ;;                                 0247.000
                                                                                     0248.000
*** DECODE FUNCTION PATTERN ***                                                       0249.000
[,F1]     'DECODE(' ( '#'@CONTROL#6 : #.(PG) ) ')' , ;;                              0250.000
                                                                                     0251.000
*** CONTROL FIELD PATTERNS ***                                                       0252.000
[CONTROL,F1]7↑54 'PUSHJ' , ;;                                                        0253.000
[CONTROL,F1]3↑54 'DECRN' , ;;                                                        0254.000
                                                                                     0255.000
*** X-ORDERS ***                                                                     0256.000
*** U-ORDERS ***                                                                     0257.000
*** SECOND PATTERN ALTERNATIVES ARE THE U-ORDERS ***                                 0258.000
*** DETERMINATION OF X/U ORDER DEFINITION IS MADE BY CPU ***                         0259.000
***  HARDWARE AND THE ENABLE U-ORDER F/F. ***                                        0260.000
[,F1]  ( %FAIL : ( 'FRCWORD' : 'RUN' ) : 'HALT' : 'RESETIO' :                        0261.000
       ( 'RLKCAR8' : 'RSTPROTV' ) : ( 'TOGRHF' : 'LDMAP' ) :                         0262.000
       ( 'SDEST' : 'RDMAP' ) : ( 'I1TOTO' : 'LDWCS' ) :                              0263.000
       ( 'PCTOMAR' : 'RDWCS' ) : ( 'SHIFTIO' : 'LDSTOP' ) :                          0264.000
         'RDLOCSTR' : 'FIXEXP' : 'SETRHF' : 'OTHERRANK' )@XU#$. , ; ;                0265.000
                                                                                     0266.000
*** Y-ORDERS ***                                                                     0267.000
[,F1]  ( %FAIL : %FAIL : %FAIL : 'FILLEXP' : %FAIL : 'SETCAR' :                      0268.000
       'USECAR' : 'WRPMAP' : 'RDLSW' :  'RSTFP'  : 'TSFILL' : 'INCRN' :              0269.000
       'RSTRHF' : %FAIL : %FAIL : 'CLKDIV' : 'SELSPARE' : 'RESETFF' :                0270.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
                'SAVESIGN' : 'RSTTPUFLG' : 'UPACK' : 'CLRTO' : 'RSTAEXP' : 'PFPAR' 0271.000
                 : %FAIL : 'FETCHPC'@SHC#8 : %FAIL : %FAIL : ( ( 'RDT' : 'ICT'     0272.000
                  )@SHC#$. : 'RSTX'@SHC#4 : ( 'READ' : 'RDLOCK' )@SHC#8+2*$. ) :   0273.000
                  'FETCHV'@SHC#8 : ( ( 'WDOT' : 'AICT' )@SHC#$. :                  0274.000
                  ( 'ARSTX' : 'WRITE' )@SHC#4+4*$. ) : %FAIL )@YO#$. , ; ;         0275.000
                                                                                   0276.000
[NOP.F,F1]$DFV '*NOP' , ;;                                                         0277.000
                                                                                   0278.000
*** CONDITIONAL ORDERS BY THEMSELVES ***                                          0279.000
[,F1]  @CO , ;;                                                                    0280.000
                                                                                   0281.000
*** SCRATCH MEMORY AND WCS READ PATTERNS ***                                      0282.000
[,F1]  'SCRATCH('@XO#@A ( #.(BLIT) : @AMUX )@D#4  ')' " '=' @BMUX , ; ;          0283.000
[,F1]  'S'@D#1 " '=' " ( 'SCRATCH('@XO#@A ( #.(BLIT) : @AMUX ) ')' :             0284.000
       'WCS(S)'@XO#6 ) , ; ;                                                       0285.000
                                                                                   0286.000
*** SHIFT FUNCTION PATTERNS ***                                                   0287.000
@SHFCOD 'SLL' : 'SLC' : 'SLLD' : 'SLCD' :                                          0288.000
        %FAIL : %FAIL : 'SLAD' : 'SLA' :                                           0289.000
        ( 'SRLD' : 'SRL' ) : ( 'SRAD' : 'SRA' ) : %FAIL : %FAIL :                  0290.000
        'SRC' : 'SRCD' ;                                                           0291.000
[,F1]  ( 'SHIFTS('@YO#1 : ( 'SHIFTDI(' : 'SHIFTD('@YO#1 )@XO#3 )                  0292.000
       ( '#'@CONTROL#2 : @SHFCOD@SHC#$. ) ')' , ; ;                                0293.000
                                                                                   0294.000
*** NIBBLE SHIFT I PATTERN ***                                                    0295.000
[,F1]    ('TNIBL'@SHC#0 : 'TNIBR'@SHC#8 )@XO#2 , ;;                               0296.000
                                                                                   0297.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7
SEL 32/75 CPU
PC MICROWORD

```
                *** ALU FUNCTION PATTERNS ***                                   0298.000
                @INC.DEC @AMUX " ( '-1' : '+1' )@ALUF#$.*2+4 ;                  0299.000
                @NOT.ALU '%' " @AMUX " ( ':' : '&' )@ALUF#$.*9+2 " @BMUX ;      0300.000
                @ALU   @AMUX " ( '+' : %FAIL : '-' : %FAIL : ':%' : '!' :       0301.000
                       %FAIL : ':' : %FAIL : '&%' : '&' : %FAIL :               0302.000
                       '#' )@ALUF#$.+3 " @BMUX ;                                0303.000
                @NOT.LIT '%' " ( @SAMUX@ALUF#9 : @BMUX.NAMES@ALUF#1 :           0304.000
                       ( #.(LIT)A )@ALUF#9 ) ;                                  0305.000
                @LITP   @SAMUX@ALUF#@E : @BMUX.NAMES@ALUF#0 :                   0306.000
                       ( #.(LIT)A )@ALUF#@E ;                                   0307.000
                [,F1] @DEST ( @INC.DEC : @NOT.ALU : @ALU : @NOT.LIT : @LITP) , ; ;  0308.000
                                                                                0309.000
                *** INDEPENDENT B MUX ACCESS ***                                0310.000
                [,F1]  'BMUX' " '=' @BMUX.NAMES , ; ;                           0311.000
                                                                                0312.000
                *** FLOATING POINT VECTORED JUMP PATTERNS ***                   0313.000
                @FPJUMP  ( ( %FAIL : '*JUMPNV' : '*JUMPDV' : '*JUMPAV'          0314.000
                        )@CONTROL#$. )@SC#7 ;                                   0315.000
                @FPORD1  ( %FAIL : 'RND' : 'CORRMNG' : 'DELSHF' : 'OVFMOPE7' :  0316.000
                       'COMPLAN' : 'COMPLAP' : 'COMPLAF' : %FAIL )@FPORDF1#$.  ;  0317.000
                @FPORD2  ( 'NOR1' : 'NOR2' : 'CORR7ERO' : 'NOR4' : 'DSFPW' : 'ASFPW' :  0318.000
                       'MASK' : 'PLUSONE' : %FAIL : 'FP.RST' )@FPOPDF2#$.  ;    0319.000
                [,F1]  @FPORD1@FPGRP1#1 : @FPORD2@FPGRP2#7 : @FPJUMP@FPVJMP#@F , ; ;  0320.000
                                                                                0321.000
                *** FLOATING POINT SPECIAL SHIFT PATTERNS WITH ALU & MPY ***     0322.000
                @STYPE " ( 'SRA' : 'SLA' : 'SRL' : 'SLL' )@FSHFTYP#$. " ;       0323.000
                @FPSD  ( 'RA'@SPDEST#0 : 'RB'@SPDEST#1 ) " '=' " ;             0324.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 5.0 - 79APR27
SEL 32/75 CPU
PC MICROWORD

```
@SPSHFT ( 'SPSHIFT(' " 'RD' " '(' )@FDMUX#6 @STYPE ',' "                    0325.000
          ( ( 'HWDCNT' : %FAIL : 'NORMCNT' : 'BITCNT' )@FCNTSEL#$. :        0326.000
          #.(FPCNT)  %MATCH@FCNTSEL#1 ) " ')' " ')'  ;                      0327.000
[,F11   ( @SPSHFT " ',' " @FPSD ( '0'@FADDR#0 : ( 'RA'.FAMUX " ( '+' :     0328.000
        '-' )@FADDR#$.+1                                                    0329.000
         " ( 'RB'.FBMUX : 'RA'.FBMUX : 'RP'.FBMUX ) ) ) ( ( " ',' "         0330.000
         'MPY(' " ( '0'@SPMPY#1 : #.(FPCNT) %MATCH@SPMPY#2 )                0331.000
         " ')' ) : %MATCH ) [ , " ( 'MPYLOOP' : 'DIVLOOP' )@AEAI#$.+1 ]     0332.000
         , ; ;                                                              0333.000
                                                                            0334.000
*** FLOATING POINT, CP FILE TO D REG PATTERN ***                            0335.000
[,F1]   'RD' " '=' " ( 'FR(' #.(FR)=$.FV:6↑4 ')' : 'R('                     0336.000
        ( @REGS : #.(BR)RF ) ')' )@R#0 " ( 'CPSOB'@FDMUX#1 :                0337.000
        ( 'CPAOB'@FDMUX#3 )@FADDR#0 ) , ; ;                                 0338.000
                                                                            0339.000
*** FLOATING POINT FRACTION GENERATE ZEROS FUNCTION PATTERNS ***            0340.000
@GEN0   '0'@FADDR#@0   ;                                                    0341.000
                                                                            0342.000
*** FLOATING POINT FRACTION SHIFT PATTERNS ***                              0343.000
@FSHFT " ( 'RB'.FBMUX : 'RA'.FBMUX : 'RP'.FBMUX :                           0344.000
          'RD'@FSIB#3 )@FADDR#3 ;                                           0345.000
                                                                            0346.000
@FSHIFT 'SHIFT(' @FSHFT " '(' @STYPE ',' "  ( ( 'HWDCNT' :                  0347.000
        %FAIL : 'NORMCNT' :                                                 0348.000
        'BITCNT' )@FCNTSEL#$. : #.(FPCNT) %MATCH@FCNTSEL#1 )                0349.000
        " ')' " ')'  ;                                                      0350.000
                                                                            0351.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SEL 32/75 CPU

PC MICROWORD

```
                *** FLOATING POINT DELSHF LEFT PATTERN ***                          0352.000
[,F1]   @FPFD 'SHIFT(' " ( 'RD'@FSIB#3 )@FADDR#3 " '(' "                           0353.000
        'DFLSHFL'@FPCNT#@30 " ')' " ')' , ; ;                                       0354.000
                                                                                    0355.000
                *** FLOATING POINT FRACTION MULTIPLY PATTERNS ***                   0356.000
[,F1]   'MPY(' " ( 'D'@FDMUX#5 : #.(FPCNT) %MATCH@FDMUX#4 )                         0357.000
        " ')' , ; ;                                                                 0358.000
                                                                                    0359.000
                *** FLOATING POINT FRACTION PATTERNS                                0360.000
@FPFD   (( 'NOD' : 'RA' : 'RB' : ( 'RA' " '%' " 'RD' ) : %FAIL :                    0361.000
        'RDMS' : 'RD' : ( 'RB' " '%' " 'RD' ))@FDEST#$. ) " '=' " :                 0362.000
        @DEST@FDEST#4 ;                                                             0363.000
@AFMUX 'RA'.FAMUX ;                                                                 0364.000
@BFMUX ( 'RB'.FBMUX : 'RA'.FBMUX : 'RP'.FBMUX ) ;                                   0365.000
@FALU   @AFMUX " ( '+' : '-' )@FADDR#$.+1 " @BFMUX ;                                0366.000
[,F11]  @FPFD   ( @FALU : @GEN0 : @FSHIFT ) , ; ;                                   0367.000
                                                                                    0368.000
                *** FLOATING POINT EXPONENT PATTERNS                                0369.000
@FPFD   ( 'NOD' : 'RBE' : 'RAE' : 'RXE' )@EDEST#$. " '=' " ;                        0370.000
@AFMUX ( 'RAE'.AEAI : 'RBE'.AEAI : 'EXPK'.AEAI : 'RMEXP'.AEAI ) ;                   0371.000
@BEMUX ( ( 'RDEXP' : 'NORCNT' )@BEAI#0 : 'RBF'.BEAT : 'RAE'.BEAI ) ;                0372.000
[,F1]   @FPFD  ( ( @AEMUX " ( '+' : '-' )@EA#$. " @BEMUX ) :                       0373.000
        ( 'RMEXP'.AEAI ) , ; ;                                                      0374.000
```

SYSTEMS MICROCODE ASSEMBLER - REVISION 3.0 - 79APR27

SFL 32/75 CPU
PC TSMAB+DR YX PCH+BDM+BAD ADDR

0375.000

0376.000

$END

E R R O R M E S S A G E S & S T A T I S T I C S

```
SYMBOL TABLE OCCUPIED   4906 BYTES (0132A HEX)
WORKSPACE STACK USED   23780 BYTES (05CE4 HEX)
UNUSED CORE SPACE WAS  41714 BYTES (0A2F2 HEX)
MAXIMUM PATTERN FAILS WAS      0
NUMBER OF CROSS REFERENCE ITEMS WAS    996
```

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

!	STRING	REFERENCED AT :	301.000							
#	STRING	REFERENCED AT :	250.000	293.000	303.000					
$	VARIABLE	REFERENCED AT :	93.000	94.000	95.000	96.000	97.000	98.000		
$.	VARIABLE	REFERENCED AT :	149.000	150.000	154.000	174.000	180.000	185.000	191.000	195.000
		197.000	199.000	201.000	203.000	211.000	213.000	216.000	218.000	221.000	223.000
		230.000	232.000	239.000	241.000	242.000	243.000	265.000	273.000	273.000	274.000
		275.000	275.000	293.000	299.000	300.000	303.000	315.000	317.000	319.000	323.000
		326.000	329.000	332.000	349.000	362.000	366.000	370.000	373.000		
$.$	VARIABLE	REFERENCED AT :	166.000							
$.FV	VARIABLE	REFERENCED AT :	151.000	153.000	159.000	163.000	336.000			
$DFV	VARIABLE	REFERENCED AT :	7.000	277.000						
$ER	UNDEF	REFERENCED AT :	5.000	151.000	159.000	163.000	241.000			
$XC	UNDEF	REFERENCED AT :	5.000	237.000	239.000	239.000	239.000			
$XT	UNDEF	REFERENCED AT :	5.000	195.000	195.000	197.000	197.000	199.000	199.000	201.000
		201.000	203.000	203.000	242.000					
%	STRING	REFERENCED AT :	237.000	241.000	242.000	243.000	300.000	304.000		
%BLKTIMEOUT	STRING	REFERENCED AT :	183.000							
%WCS	STRING	REFERENCED AT :	183.000							
&	STRING	REFERENCED AT :	300.000	302.000	361.000	362.000				
&%	STRING	REFERENCED AT :	302.000							
*	STRING	REFERENCED AT :	244.000							
*JUMPAV	STRING	REFERENCED AT :	314.000							
*JUMPDV	STRING	REFERENCED AT :	314.000							
*JUMPNV	STRING	REFERENCED AT :	314.000							
*JUMPS	STRING	REFERENCED AT :	238.000							
*NOP	STRING	REFERENCED AT :	277.000							
+	STRING	REFERENCED AT :	301.000	328.000	366.000	373.000				
+1	STRING	REFERENCED AT :	299.000							
-	STRING	REFERENCED AT :	301.000	329.000	366.000	373.000				
-1	STRING	REFERENCED AT :	299.000							

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 32/75 C P U

0	STRING	REFERENCED AT :	328.000	341.000					
:	STRING	REFERENCED AT :	204.000	205.000	206.000	207.000	300.000	302.000	
:%	STRING	REFERENCED AT :	301.000						
=	STRING	REFERENCED AT :	155.000	283.000	284.000	311.000	324.000	336.000	362.000	370.000
A	(P)FIELD	REFERENCED AT :	21.000	113.000	114.000	115.000	116.000	117.000	118.000	159.000
		160.000 160.000	305.000	307.000						
ABSDI	STRING	REFERENCED AT :	229.000							
ABST	STRING	REFERENCED AT :	229.000							
ADDRSTOP	STRING	REFERENCED AT :	203.000							
AEAI	(P)FIELD	REFERENCED AT :	64.000	138.000	139.000	140.000	141.000	332.000	371.000	371.000
		371.000 371.000	374.000							
AEMUX	PATTERN	REFERENCED AT :	371.000	373.000						
AEXP	STRING	REFERENCED AT :	199.000	232.000						
AFMUX	PATTERN	REFERENCED AT :	364.000	366.000						
AICT	STRING	REFERENCED AT :	274.000							
AL	STRING	REFERENCED AT :	231.000							
ALTERFF1	PATTERN	REFERENCED AT :	211.000	230.000						
ALTERFF2	PATTERN	REFERENCED AT :	216.000	230.000						
ALTERFF3	PATTERN	REFERENCED AT :	221.000	231.000						
ALU	PATTERN	REFERENCED AT :	301.000	308.000						
ALU4-77	STRING	REFERENCED AT :	189.000							
ALUF	(P)FIELD	REFERENCED AT :	30.000	299.000	300.000	303.000	304.000	304.000	305.000	306.000
		306.000 307.000								
ALUNEG	STRING	REFERENCED AT :	190.000							
ALUNEGW	STRING	REFERENCED AT :	177.000							
ALUZ	STRING	REFERENCED AT :	173.000							
AMUX	PATTERN	REFERENCED AT :	159.000	283.000	284.000	299.000	300.000	301.000		
AR	(P)FIELD	REFERENCED AT :	78.000	160.000						
ARSTX	STRING	REFERENCED AT :	275.000							

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

ASEPW	STRING	REFERENCED AT :	318.000							
AYO	(P)FIELD	REFERENCED AT :	24.000	156.000						
B	(P)FIELD	REFERENCED AT :	29.000	121.000	122.000	123.000	124.000	125.000	126.000	127.000
			166.000	167.000	168.000	168.000	168.000	168.000	169.000	169.000	169.000	337.000
BADSCALE	STRING	REFERENCED AT :	196.000							
BEAI	(P)FIELD	REFERENCED AT :	63.000	144.000	145.000	372.000	372.000	372.000		
BEMUX	PATTERN	REFERENCED AT :	372.000	373.000						
BFMUX	PATTERN	REFERENCED AT :	365.000	366.000						
BIBUSY	STRING	REFERENCED AT :	199.000							
BIT	STRING	REFERENCED AT :	232.000							
BITCNT	STRING	REFERENCED AT :	326.000	349.000						
BLIT	(P)FIELD	REFERENCED AT :	28.000	90.000	90.000	283.000	284.000			
BLKCARR	STRING	REFERENCED AT :	262.000							
BMG	REGISTER	REFERENCED AT :	118.000							
BMGMR	STRING	REFERENCED AT :	156.000							
BMOD	PATTERN	REFERENCED AT :	161.000	165.000	165.000	167.000	169.000			
BMUX	STRING	REFERENCED AT :	311.000							
BMUX	PATTERN	REFERENCED AT :	167.000	283.000	300.000	303.000				
BMUX.NAMES	PATTERN	REFERENCED AT :	168.000	304.000	306.000	311.000				
BMUX00	STRING	REFERENCED AT :	178.000							
BMUX16	STRING	REFERENCED AT :	179.000							
BMUX17	STRING	REFERENCED AT :	179.000							
BMUX18	STRING	REFERENCED AT :	179.000							
BMUX19	STRING	REFERENCED AT :	179.000							
BMUXR	PATTERN	REFERENCED AT :	163.000	167.000	168.000					
BR	(P)FIELD	REFERENCED AT :	79.000	164.000	337.000					
BRANCH	STRING	REFERENCED AT :	245.000							
BRANCH	(P)FIELD	REFERENCED AT :	41.000	95.000	95.000	98.000	109.000	245.000		

M I C R O C O D E C R O S S - R E F F R F N C E L I S T

S E L 3 2 / 7 5 C P U

BYTE	STRING	REFERENCED AT :	180.000						
C	STRING	REFERENCED AT :	232.000						
CC	(P)FIELD	REFERENCED AT :	19.000	232.000	233.000				
CCTEST	STRING	REFERENCED AT :	189.000						
CDECRN	STRING	REFERENCED AT :	230.000						
CLDNU	STRING	REFERENCED AT :	229.000						
CLKDIV	STRING	REFERENCED AT :	270.000						
CLRS	STRING	REFERENCED AT :	228.000						
CLRSYSR	STRING	REFERENCED AT :	227.000						
CLRTO	STRING	REFERENCED AT :	271.000						
CO	PATTERN	REFERENCED AT :	227.000	237.000	242.000	243.000	243.000	244.000	280.000
COMPLAF	STRING	REFERENCED AT :	317.000						
COMPLAN	STRING	REFERENCED AT :	317.000						
COMPLAP	STRING	REFERENCED AT :	317.000						
CONDO	(P)FIELD	REFERENCED AT :	16.000	213.000	218.000	224.000	230.000	233.000	235.000
CONTROL	(P)FIELD	REFERENCED AT : 315.000	51.000	231.000	239.000	247.000	250.000	253.000	254.000 293.000
CORKMNG	STRING	REFERENCED AT :	316.000						
CORRZEROW	STRING	REFERENCED AT :	318.000						
CPAOR	STRING	REFERENCED AT :	338.000						
CPSOR	STRING	REFERENCED AT :	337.000						
D	STRING	REFERENCED AT :	232.000	351.000	357.000				
D	(P)FIELD	REFERENCED AT :	11.000	149.000	150.000	151.000	152.000	283.000	284.000
DECODE(	STRING	REFERENCED AT :	250.000						
DECRN	STRING	REFERENCED AT :	254.000						
DELSHF	STRING	REFERENCED AT :	316.000						
DELSHFL	STRING	REFERENCED AT :	354.000						
DEST	PATTERN	REFERENCED AT :	151.000	308.000	363.000				

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

DF	(P)FIELD	REFERENCED AT :	12.000	152.000	152.000	152.000	153.000	153.000	153.000
DI	STRING	REFERENCED AT :	148.000	168.000					
DI	REGISTER	REFERENCED AT :	125.000						
DINTRA	STRING	REFERENCED AT :	212.000						
DIS.BLK.TIMEOUT	STRING	REFERENCED AT :	223.000						
DIVIDE	STRING	REFERENCED AT :	229.000						
DIVLOOP	STRING	REFERENCED AT :	332.000						
DIVLSW	STRING	REFERENCED AT :	238.000						
DIVMSW	STRING	REFERENCED AT :	238.000						
DIX	STRING	REFERENCED AT :	153.000	158.000					
DPEEF	STRING	REFERENCED AT :	212.000						
DREG	PATTERN	REFERENCED AT :	148.000	154.000					
DSFPW	STRING	REFERENCED AT :	318.000						
DWAIT	STRING	REFERENCED AT :	213.000						
DWORD	STRING	REFERENCED AT :	180.000						
E	STRING	REFERENCED AT :	232.000						
FA	(P)FIELD	REFERENCED AT :	62.000	373.000					
FDEST	(P)FIELD	REFERENCED AT :	65.000	370.000					
ELSA	STRING	REFERENCED AT :	184.000	222.000					
ENAINTFF	STRING	REFERENCED AT :	213.000						
ENASORD	STRING	REFERENCED AT :	222.000						
ENATRMT	STRING	REFERENCED AT :	217.000						
ENAUORD	STRING	REFERENCED AT :	216.000						
ENBL.AFXP	STRING	REFERENCED AT :	184.000	222.000					
ENBL55	STRING	REFERENCED AT :	194.000						
ETR0	(P)FIELD	REFERENCED AT :	34.000	203.000					
ETR1	(P)FIELD	REFERENCED AT :	35.000	201.000					

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

ETB2	(P)FIELD	REFERENCED AT :	36.000	199.000					
ETB3	(P)FIELD	REFERENCED AT :	37.000	197.000					
ETB4	(P)FIELD	REFERENCED AT :	33.000	195.000					
EXFF	STRING	REFERENCED AT :	211.000						
EXFLAG	STRING	REFERENCED AT :	200.000	212.000					
EXPK	STRING	REFERENCED AT :	371.000						
EXPK	REGISTER	REFERENCED AT :	140.000						
EXTG	STRING	REFERENCED AT :	203.000						
EXTL	STRING	REFERENCED AT :	188.000						
EXTLW	STRING	REFERENCED AT :	177.000						
EXTMAPERR	STRING	REFERENCED AT :	185.000						
EXTST	(P)FIELD	REFERENCED AT :	38.000	242.000					
F1	FORMAT	REFERENCED AT :	106.000	237.000	241.000	250.000	253.000	254.000	261.000	268.000
			277.000	280.000	283.000	284.000	292.000	296.000	308.000	311.000	320.000	328.000
			336.000	353.000	357.000	367.000	373.000					
FADDR	(P)FIELD	REFERENCED AT :	54.000	328.000	329.000	338.000	341.000	345.000	353.000	366.000		
FALSE	STRING	REFERENCED AT :	174.000									
FALU	PATTERN	REFERENCED AT :	366.000	367.000								
FAMUX	(P)FIELD	REFERENCED AT :	55.000	130.000	328.000	364.000						
FBMUX	(P)FIELD	REFERENCED AT :	56.000	133.000	134.000	135.000	330.000	330.000	330.000	344.000
			344.000	344.000	365.000	365.000	365.000					
FC1V	STRING	REFERENCED AT :	191.000									
FCNTSEL	(P)FIELD	REFERENCED AT :	71.000	326.000	327.000	349.000	349.000					
FDEST	(P)FIELD	REFERENCED AT :	58.000	362.000	363.000							
FDMUX	(P)FIELD	REFERENCED AT :	61.000	325.000	337.000	338.000	357.000	357.000				
FETCHPC	STRING	REFERENCED AT :	272.000									
FETCHV	STRING	REFERENCED AT :	274.000									
FF1	(P)FIELD	REFERENCED AT :	45.000	213.000								
FF2	(P)FIELD	REFERENCED AT :	47.000	218.000								

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

FF3	(P)FIELD	REFERENCED AT :	49.000	223.000						
FFINT	STRING	REFERENCED AT :	200.000							
FFR	(P)FIELD	REFERENCED AT :	23.000	81.000						
FFRUN	STRING	REFERENCED AT :	177.000							
FFSYSR	STRING	REFERENCED AT :	194.000							
FILLEXP	STRING	REFERENCED AT :	268.000							
FIXEXP	STRING	REFERENCED AT :	265.000							
FLAG	STRING	REFERENCED AT :	201.000	218.000						
FORCFZ	STRING	REFERENCED AT :	227.000							
FP.RST	STRING	REFERENCED AT :	319.000							
FPCNT	(P)FIELD	REFERENCED AT :	73.000	102.000	102.000	103.000	103.000	327.000	331.000	349.000
		354.000	357.000							
FPDWORD	STRING	REFERENCED AT :	223.000							
FPED	PATTERN	REFERENCED AT :	370.000	373.000						
FPFD	PATTERN	REFERENCED AT :	353.000	361.000	367.000					
FPGRP1	(P)FIELD	REFERENCED AT :	68.000	320.000						
FPGRP2	(P)FIELD	REFERENCED AT :	69.000	320.000						
FPJUMP	PATTERN	REFERENCED AT :	314.000	320.000						
FPORD1	PATTERN	REFERENCED AT :	316.000	320.000						
FPORD2	PATTERN	REFERENCED AT :	318.000	320.000						
FPORDF1	(P)FIELD	REFERENCED AT :	75.000	317.000						
FPORDF2	(P)FIELD	REFERENCED AT :	74.000	319.000						
FPSD	PATTERN	REFERENCED AT :	324.000	328.000						
FPU	STRING	REFERENCED AT :	232.000							
FPVJMP	(P)FIELD	REFERENCED AT :	70.000	320.000						
FR	(P)FIELD	REFERENCED AT :	22.000	81.000	151.000	157.000	157.000	157.000	157.000	157.000
		158.000	158.000	159.000	163.000	336.000				
FRF	STRING	REFERENCED AT :	151.000	159.000	163.000	336.000				
FRCWORD	STRING	REFERENCED AT :	261.000							

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

FSHFT	PATTERN	REFERENCED AT :	344.000	347.000				
FSHFTYP	(P)FIELD	REFERENCED AT :	72.000	323.000				
FSHIFT	PATTERN	REFERENCED AT :	347.000	367.000				
FSIB	(P)FIELD	REFERENCED AT :	57.000	345.000	353.000			
FULLMAR	STRING	REFERENCED AT :	150.000					
GEN0	PATTERN	REFERENCED AT :	341.000	367.000				
GO TO	STRING	REFERENCED AT :	246.000					
GOTO	STRING	REFERENCED AT :	246.000					
GOTO	(P)FIELD	REFERENCED AT :	96.000	97.000	98.000	99.000	109.000	246.000	247.000
HALT	STRING	REFERENCED AT :	261.000					
HIREG	STRING	REFERENCED AT :	196.000	211.000				
HOP	STRING	REFERENCED AT :	245.000					
HOP	(P)FIELD	REFERENCED AT :	39.000	93.000	93.000	96.000	109.000	245.000	
HWDCNT	STRING	REFERENCED AT :	326.000	347.000				
HWORD	STRING	REFERENCED AT :	180.000					
HWS	STRING	REFERENCED AT :	162.000					
HWS	PATTERN	REFERENCED AT :	162.000	165.000	165.000			
I0	STRING	REFERENCED AT :	168.000					
I0	REGISTER	REFERENCED AT :	124.000					
I1	STRING	REFERENCED AT :	148.000					
I1TOI0	STRING	REFERENCED AT :	263.000					
ICT	STRING	REFERENCED AT :	272.000					
IF	STRING	REFERENCED AT :	237.000	241.000				
IGNSTOP	STRING	REFERENCED AT :	227.000					
INC.DEC	PATTERN	REFERENCED AT :	299.000	308.000				
INCRN	STRING	REFERENCED AT :	269.000					
INDIR	STRING	REFERENCED AT :	191.000					

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

INSTMIIER	STRING	REFERENCED AT :	202.000					
INSTNORESP	STRING	REFERENCED AT :	196.000					
INSTTIMEOUT	STRING	REFERENCED AT :	200.000					
INTLVL	STRING	REFERENCED AT :	169.000					
INTLVL	REGISTER	REFERENCED AT :	126.000					
INTRENA	STRING	REFERENCED AT :	201.000					
IOCHBUSY	STRING	REFERENCED AT :	198.000					
IONORESP	STRING	REFERENCED AT :	196.000					
IORESPRDY	STRING	REFERENCED AT :	196.000					
IORETRY	STRING	REFERENCED AT :	202.000					
IOTIMEOUT	STRING	REFERENCED AT :	200.000					
IPLSW	STRING	REFERENCED AT :	202.000					
IPU	STRING	REFERENCED AT :	183.000					
IPU.HALT	STRING	REFERENCED AT :	185.000	217.000				
IPU.START	STRING	REFERENCED AT :	183.000	217.000				
JUMPD	STRING	REFERENCED AT :	244.000					
JUMPJ	STRING	REFERENCED AT :	246.000					
JUMPZ	STRING	REFERENCED AT :	246.000					
LATERR	STRING	REFERENCED AT :	199.000					
LATERRW	STRING	REFERENCED AT :	180.000					
LATERRZ	STRING	REFERENCED AT :	189.000					
LDMAP	STRING	REFERENCED AT :	262.000					
LDSTOP	STRING	REFERENCED AT :	264.000					
LDWCS	STRING	REFERENCED AT :	263.000					
LEAP	STRING	REFERENCED AT :	245.000					
LEAP	(P)FIELD	REFERENCED AT :	40.000	94.000	94.000	97.000	109.000	245.000
LFC1V	STRING	REFERENCED AT :	178.000					
LH	STRING	REFERENCED AT :	154.000					

M I C R O C O D E C R O S S - R E F E R E N C E L I S T

S E L 3 2 / 7 5 C P U

LINK	STRING	REFERENCED AT :	247.000						
LIT	(P)FIELD	REFERENCED AT :	26.000	82.000	82.000	83.000	84.000	85.000	86.000	87.000
			88.000	89.000	160.000	305.000	307.000			
LITP	PATTERN	REFERENCED AT :	306.000	308.000						
LLIT	(P)FIELD	REFERENCED AT :	27.000	83.000	84.000	85.000	86.000	87.000	88.000	89.000
LOCSTORE	STRING	REFERENCED AT :	148.000							
MAPINVALID	STRING	REFERENCED AT :	195.000							
MAPMODE	STRING	REFERENCED AT :	195.000	217.000						
MAR	STRING	REFERENCED AT :	148.000	168.000						
MAR	REGISTER	REFERENCED AT :	122.000							
MARIX	STRING	REFERENCED AT :	149.000							
MASK	STRING	REFERENCED AT :	319.000							
MODE75	STRING	REFERENCED AT :	188.000	221.000						
MODE75S	STRING	REFERENCED AT :	185.000							
MPROM	STRING	REFERENCED AT :	236.000							
MPY	STRING	REFERENCED AT :	229.000							
MPY(	STRING	REFERENCED AT :	331.000	357.000						
MPYLOOP	STRING	REFERENCED AT :	332.000							
N	STRING	REFERENCED AT :	169.000							
N	REGISTER	REFERENCED AT :	123.000							
NALU7	STRING	REFERENCED AT :	174.000							
NCTR	STRING	REFERENCED AT :	152.000	157.000						
NCTR0	STRING	REFERENCED AT :	191.000							
NCTR4	STRING	REFERENCED AT :	190.000							
NCTR7	STRING	REFERENCED AT :	190.000							
NHMPREV	STRING	REFERENCED AT :	188.000							
NL	STRING	REFERENCED AT :	150.000							
NOD	STRING	REFERENCED AT :	148.000	361.000	370.000					

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

NOEXTUNIV	STRING	REFERENCED AT :	173.000				
NOP.F	(P)FIELD	REFERENCED AT :	10.000	277.000			
NOR1	STRING	REFERENCED AT :	316.000				
NOR2	STRING	REFERENCED AT :	316.000				
NOR4	STRING	REFERENCED AT :	316.000				
NORC	STRING	REFERENCED AT :	176.000				
NORCNT	STRING	REFERENCED AT :	372.000				
NORM	STRING	REFERENCED AT :	239.000				
NORMCNT	STRING	REFERENCED AT :	326.000	348.000			
NOT.ALU	PATTERN	REFERENCED AT :	300.000	308.000			
NOT.LIT	PATTERN	REFERENCED AT :	304.000	308.000			
NU	STRING	REFERENCED AT :	150.000				
OPMIUER	STRING	REFERENCED AT :	203.000				
OPNORESP	STRING	REFERENCED AT :	197.000				
OPRNOPE	STRING	REFERENCED AT :	195.000				
OPTIMEOUT	STRING	REFERENCED AT :	201.000				
OTHERBANK	STRING	REFERENCED AT :	265.000				
OVFMOPFZ	STRING	REFERENCED AT :	316.000				
PC	STRING	REFERENCED AT :	148.000				
PCTOMAR	STRING	REFERENCED AT :	264.000				
PFPAR	STRING	REFERENCED AT :	271.000				
PG	(P)FIELD	REFERENCED AT :	44.000	91.000	91.000	92.000	92.000	250.000
PLUSONE	STRING	REFERENCED AT :	319.000				
PNLDATA	STRING	REFERENCED AT :	169.000				
PNLDATA	REGISTER	REFERENCED AT :	127.000				
PPATTN	STRING	REFERENCED AT :	202.000				
PRIV	STRING	REFERENCED AT :	212.000				

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

PRIVBIT	STRING	REFERENCED AT :	198.000						
PROTV	STRING	REFERENCED AT :	197.000						
PUSHJ	STRING	REFERENCED AT :	253.000						
PWRFAIL	STRING	REFERENCED AT :	198.000						
R	STRING	REFERENCED AT :	152.000	157.000					
R	(P)FIELD	REFERENCED AT :	20.000	80.000	153.000				
R(	STRING	REFERENCED AT :	152.000	160.000	164.000	336.000			
RA	STRING	REFERENCED AT :	324.000	328.000	330.000	344.000	361.000	361.000	364.000	365.000
RA	REGISTER	REFERENCED AT :	130.000	134.000					
RAF	STRING	REFERENCED AT :	370.000	371.000	372.000				
RAE	REGISTER	REFERENCED AT :	138.000	145.000					
RB	STRING	REFERENCED AT :	324.000	330.000	344.000	361.000	362.000	365.000	
RB	REGISTER	REFERENCED AT :	133.000						
RBE	STRING	REFERENCED AT :	370.000	371.000	372.000				
RBE	REGISTER	REFERENCED AT :	139.000	144.000					
RD	STRING	REFERENCED AT :	152.000	157.000	325.000	336.000	345.000	353.000	361.000	362.000
		362.000							
RDEXP	STRING	REFERENCED AT :	372.000						
RDLOCK	STRING	REFERENCED AT :	273.000						
RDLOCSTR	STRING	REFERENCED AT :	265.000						
RDLSW	STRING	REFERENCED AT :	269.000						
RDMAP	STRING	REFERENCED AT :	263.000						
RDMS	STRING	REFERENCED AT :	362.000						
RDT	STRING	REFERENCED AT :	272.000						
RDWCS	STRING	REFERENCED AT :	264.000						
READ	STRING	REFERENCED AT :	273.000						
REGS	PATTERN	REFERENCED AT :	157.000	160.000	164.000	337.000			
REGSEL	STRING	REFERENCED AT :	237.000						

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

REPEAT	STRING	REFERENCED AT :	238.000					
RESET(	STRING	REFERENCED AT :	211.000	216.000	221.000			
RESETFF	STRING	REFERENCED AT :	270.000					
RESETIO	STRING	REFERENCED AT :	261.000					
RF	(P)FIELD	REFERENCED AT :	13.000	78.000	79.000	160.000	164.000	337.000
RH	STRING	REFERENCED AT :	154.000					
RHFLAG	STRING	REFERENCED AT :	190.000					
RMEXP	STRING	REFERENCED AT :	371.000	374.000				
RMEXP	REGISTER	REFERENCED AT :	141.000					
RN	(P)FIELD	REFERENCED AT :	80.000	153.000				
RND	STRING	REFERENCED AT :	316.000					
RO	STRING	REFERENCED AT :	152.000	157.000				
RP	STRING	REFERENCED AT :	330.000	344.000	365.000			
RP	REGISTER	REFERENCED AT :	135.000					
RSTAEXP	STRING	REFERENCED AT :	271.000					
RSTFP	STRING	REFERENCED AT :	269.000					
RSTIPUFLG	STRING	REFERENCED AT :	271.000					
RSTPROTV	STRING	REFERENCED AT :	262.000					
RSTRHF	STRING	REFERENCED AT :	270.000					
RSTX	STRING	REFERENCED AT :	273.000					
RUN	STRING	REFERENCED AT :	261.000					
RXF	STRING	REFERENCED AT :	370.000					
S	STRING	REFERENCED AT :	148.000	153.000	157.000	232.000	233.000	284.000
S	REGISTER	REFERENCED AT :	113.000					
S.R.FF1	(P)FIELD	REFERENCED AT :	46.000	211.000				
S.R.FF2	(P)FIELD	REFERENCED AT :	48.000	216.000				
S.R.FF3	(P)FIELD	REFERENCED AT :	50.000	221.000				
SONETO	STRING	REFERENCED AT :	197.000					

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

SAMUX	PATTERN	REFERENCED AT :	156.000	160.000	304.000	306.000				
SAVESIGN	STRING	REFERENCED AT :	271.000							
SC	(P)FIELD	REFERENCED AT :	43.000	244.000	246.000	246.000	315.000			
SCALE	STRING	REFERENCED AT :	239.000							
SCRATCH(	STRING	REFERENCED AT :	283.000	284.000						
SDEST	STRING	REFERENCED AT :	263.000							
SE	STRING	REFERENCED AT :	161.000							
SELSPARE	STRING	REFERENCED AT :	270.000							
SERIAL.PANEL	STRING	REFERENCED AT :	194.000							
SET(	STRING	REFERENCED AT :	211.000	216.000	221.000					
SETAEXP	STRING	REFERENCED AT :	230.000							
SETCAR	STRING	REFERENCED AT :	268.000							
SETCC(	STRING	REFERENCED AT :	231.000							
SETCC(#)	STRING	REFERENCED AT :	231.000							
SETRHF	STRING	REFERENCED AT :	265.000							
SETXCC(	STRING	REFERENCED AT :	233.000							
SHC	(P)FIELD	REFERENCED AT :	18.000	272.000	273.000	273.000	273.000	274.000	274.000	275.000
			293.000	296.000	296.000					
SHFCOD	PATTERN	REFERENCED AT :	288.000	293.000						
SHIFT	(P)FIELD	REFERENCED AT :	17.000							
SHIFT(	STRING	REFERENCED AT :	347.000	353.000						
SHIFTD(	STRING	REFERENCED AT :	292.000							
SHIFTDT(	STRING	REFERENCED AT :	292.000							
SHIFTIO	STRING	REFERENCED AT :	264.000							
SHIFTS(	STRING	REFERENCED AT :	292.000							
SIGNSAVE	STRING	REFERENCED AT :	177.000							
SIGNSAVE7	STRING	REFERENCED AT :	189.000							
SLA	STRING	REFERENCED AT :	289.000	323.000						

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

SLAD	STRING	REFERENCED AT :	289.000					
SLC	STRING	REFERENCED AT :	288.000					
SLCD	STRING	REFERENCED AT :	288.000					
SLEFT	REGISTER	REFERENCED AT :	114.000					
SLL	STRING	REFERENCED AT :	286.000	323.000				
SLLD	STRING	REFERENCED AT :	288.000					
SNTBL	REGISTER	REFERENCED AT :	115.000					
SNTBR	REGISTER	REFERENCED AT :	116.000					
SPDEST	(P)FIELD	REFERENCED AT :	59.000	324.000	324.000			
SPMPY	(P)FIELD	REFERENCED AT :	60.000	331.000	331.000			
SPSHFT	PATTERN	REFERENCED AT :	325.000	328.000				
SPSHIFT(	STRING	REFERENCED AT :	325.000					
SRA	STRING	REFERENCED AT :	290.000	323.000				
SRAD	STRING	REFERENCED AT :	290.000					
SRC	STRING	REFERENCED AT :	291.000					
SRCD	STRING	REFERENCED AT :	291.000					
SRL	STRING	REFERENCED AT :	290.000	323.000				
SRLD	STRING	REFERENCED AT :	290.000					
STATUS	REGISTER	REFERENCED AT :	117.000					
STEST	PATTERN	REFERENCED AT :	193.000	244.000				
STYPE	PATTERN	REFERENCED AT :	323.000	325.000	347.000			
T	STRING	REFERENCED AT :	150.000	168.000				
T	REGISTER	REFERENCED AT :	121.000					
T03SIG	STRING	REFERENCED AT :	198.000					
TEST	PATTERN	REFERENCED AT :	172.000	241.000				
TEST0	PATTERN	REFERENCED AT :	202.000	204.000	205.000	206.000	207.000	208.000
TEST1	PATTERN	REFERENCED AT :	200.000	204.000	205.000	206.000	207.000	208.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

TEST2	PATTERN	REFERENCED AT :	198.000	204.000	205.000	206.000	207.000		
TEST3	PATTERN	REFERENCED AT :	196.000	204.000	205.000	206.000	207.000	208.000	208.000
TEST4	PATTERN	REFERENCED AT :	194.000	204.000	205.000	206.000	207.000	208.000	
TESTF	(P)FIELD	REFERENCED AT :	31.000	174.000	239.000	241.000	242.000	243.000	244.000
TNIBL	STRING	REFERENCED AT :	296.000						
TNIBR	STRING	REFERENCED AT :	296.000						
TOGRHF	STRING	REFERENCED AT :	262.000						
TRACE	STRING	REFERENCED AT :	202.000						
TRACEFF	STRING	REFERENCED AT :	212.000						
TRUE	STRING	REFERENCED AT :	188.000						
TSFILL	STRING	REFERENCED AT :	269.000						
UARTDAV	STRING	REFERENCED AT :	194.000	217.000					
UARTDS	STRING	REFERENCED AT :	216.000						
UARTERR	STRING	REFERENCED AT :	194.000						
UARTTBMT	STRING	REFERENCED AT :	200.000						
UNBLOCK	STRING	REFERENCED AT :	184.000	221.000					
UPACK	STRING	REFERENCED AT :	271.000						
UPREQ	STRING	REFERENCED AT :	198.000						
USECAR	STRING	REFERENCED AT :	269.000						
V	STRING	REFERENCED AT :	231.000						
WCS(S)	STRING	REFERENCED AT :	155.000	285.000					
WCS.BR	(P)FIELD	REFERENCED AT :	42.000	99.000	110.000				
WDOT	STRING	REFERENCED AT :	274.000						
WRITF	STRING	REFERENCED AT :	275.000						
WRPMAP	STRING	REFERENCED AT :	269.000						
WTFST	PATTERN	REFERENCED AT :	177.000	243.000					
X	STRING	REFERENCED AT :	153.000	158.000					
XO	(P)FIELD	REFERENCED AT :	14.000	155.000	161.000	161.000	180.000	185.000	265.000	283.000

MICROCODE CROSS-REFERENCE LIST

SEL 32/75 CPU

		284.000	285.000	292.000	296.000		
XTEST	PATTERN REFERENCED AT :	204.000	242.000				
Y0	(P)FIELD REFERENCED AT :	15.000	154.000	275.000	292.000	292.000	
YX0	(P)FIELD REFERENCED AT :	25.000					
ZE	STRING REFERENCED AT :	161.000					
ZTEST	PATTERN REFERENCED AT :	188.000	237.000	241.000			
ZTESTE	(P)FIELD REFERENCED AT :	32.000	191.000				

* * * * *

```
$ASSIGN1 SI=DRDEF
$OPTION 2 3
$EXECUTE MICRO
```

```
PC   MICROWORD

                    *** D-ROM DEFINTIONS FOR BURNED PROMS ON COPPER CPU ***                0001.000
                    *** 08 DECEMBER 1977 ***                                              0002.000
                          $SIZE    24                                                     0003.000
00FFF8              ADDR  $FD      3,13      13 BIT BRANCH ADDRESS STARTING IN BIT 03(20)  0004.000
                    ADDR  $PD      (ADDR)16;1,0                                           0005.000
F80000              ALU.A $FD      19,5                                                   0006.000
F00000              SC.A  $FD      20,4      SHIFT CODE                                   0007.000
070000              CC.A  $FD      16,3     CONDITION CODE                                0008.000
0F0000              CC.FP $FD      16,4     FAST FLOATING POINT CONDITION CODE            0009.000
FF0000              PAGE  $FD      16,8                                                   0010.000
                    PAGE  $PD      (PAGE)8;1,0                                            0011.000
                    F2    $FMT     4,16,8;10,0,16;18,(ALU.A);24,(SC.A);                  0012.000
                                   30,(CC.A);30,(CC.FP);36,(PAGE);42,((ADDR))            0013.000
                    (,F2) ( ( 'A' : 'A.AND.B' : 'A.AND.-B' : %FAIL : 'A.OR.B' : 'B' :    0014.000
                          'A.EOR.B' : '-A.AND.B' : 'A.OR.-B' : %FAIL : '-B' : %FAIL :    0015.000
                          %FAIL : '-A.OR.B' : %FAIL : '-A' : 'A.PLUS.1' )@ALU.A#$. :     0016.000
                          'A.PLUS.B'@ALU.A#@16 : 'A.MINUS.B'@ALU.A#@19 :                 0017.000
                          'A.MINUS.1'@ALU.A#@1F ) , ; ;                                  0018.000
                    (,F2) ( 'SLL' : 'SLC' : 'SLLD' : 'SLCD' :                            0019.000
                          %FAIL : %FAIL : 'SLAD' : 'SLA' :                               0020.000
                          ( 'SRLD' : 'SRL' ) : ( 'SRAD' : 'SRA' ) : %FAIL : %FAIL :      0021.000
                          'SRC' : 'SRCD' )@SC.A#$. , ; ;                                 0022.000
                    (,F2) 'FFPCC'@CC.FP#8 , ; ;                                          0023.000
                    (,F2) ( 'X'@PAGE#0 : 'NP=' #.(PAGE) ) , ; ;                          0024.000
                    (,F2) ( 'AL':'V':'E':'D':'BIT':'S':'AEXP':'C' )@CC.A#$.   , ;;        0025.000
                    (,F2) #.(ADDR) , ; ;                                                 0026.000
```

S Y S T E M S M I C R O C O D E A S S E M B L E R - R E V I S I O N 3.0 - 7 9 A P R 2 7

PC D(1) D(2) ALU SC CC PAGE ADDR

0027.000

$END

0028.000