MACHINE TYPE 2025 MODEL 000 CORE LOAD 1952360 *E60 MES/FCSI # SERIAL # 10302-

MASTER CONTROL MACHINE STATUS E

EC # 128266 SUM CHECK # 63CC ** SEE *E60 BCHK ** EC # TAPE USED 128267 07/130 F2 DATE PREPARED 07/13/71

FEATURES PRESENT

2542057	MEM48K	2539229	FLTPT	0854021	STGPROT	2539257	PFR
2539256	BURSTCH	0854031	TIMER	0854019	NATV2540	0854015	NATV1403
0854017	NATV2311	0854095	MULCHSET				

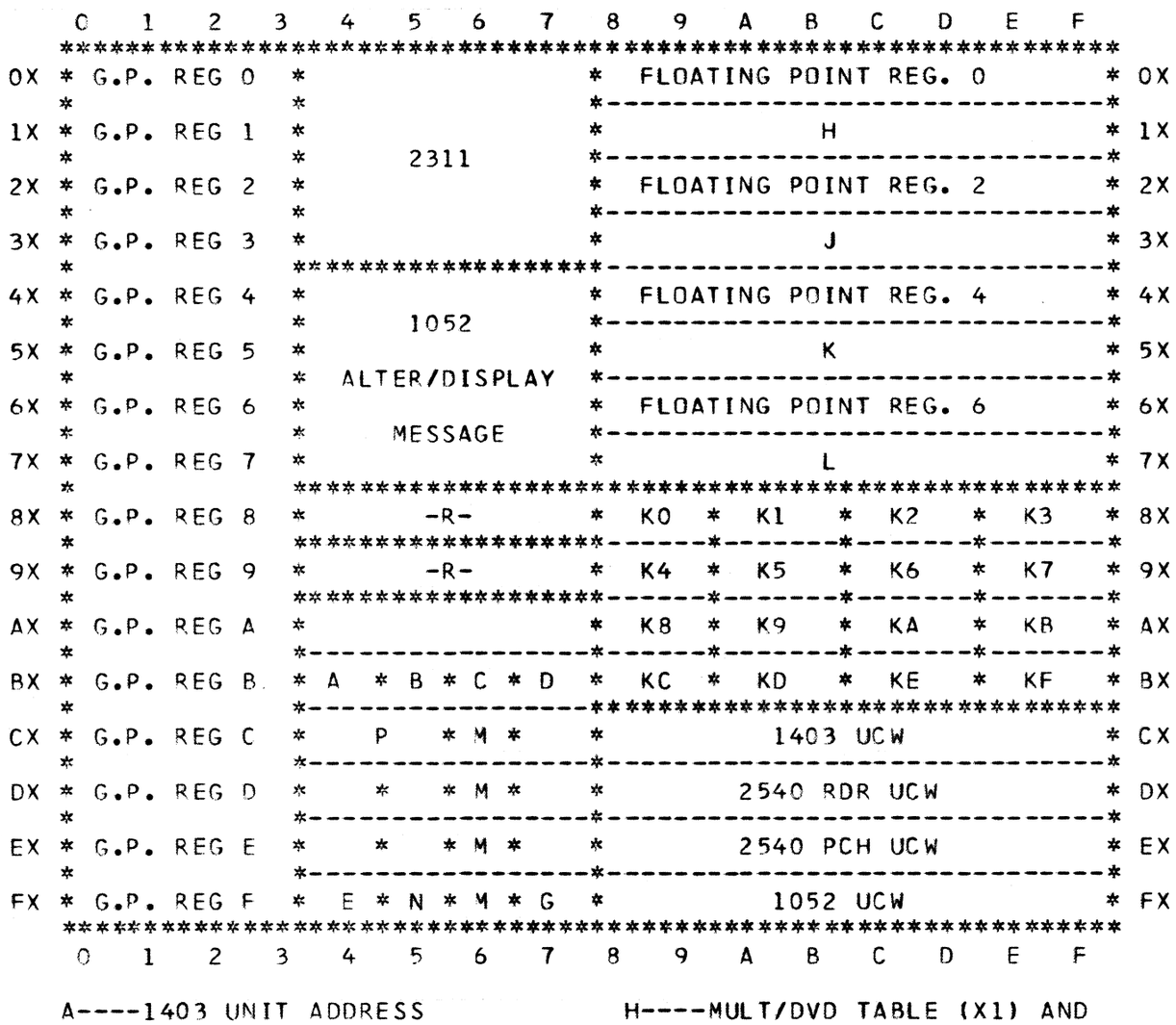
FEATURES NOT PRESENT

2542054	MEM16K	2542056	MEM32K	2539255	BYTECH	0854023	DIRCTRL
2542055	MEM24K	0854025	EXTINT	0854030	TIMER50	2532200	COMBASIC
2470709	2560EM40	2470710	CS1400	0882211	RPQMII	0882424	RPQMSC
0882447	RPQHSC	0727736	RPQPR6	0882470	RPQSUM	0882469	RPQAAR
0882376	RPQHWL	0882377	RPQBUR	0882214	RPQFAF	0882219	RPQFAC
0882225	RPQGAM						

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                  *E60 001    T                 SYSTEM 360/25 EMULATOR
                  *E60 005    *                       AAAB
                  *E60 006    *                       AAAC       LOCAL STORAGE MAP  CPU
                  *E60 008    *                       AAAD       EXTERNAL MNEMONIC DEFINITIONS
                  *E60 009    *                       AAAF       BIT SIGNIFICANCE CHARTS FOR CONTROL WORDS
                  *E60 010    *                       AAAG       MPTF ENTRY SHEETS
                  *E60 030    *                       ALDP       ALTER DISPLAY ROUTINE
                  *E60 044    *                       BCHK        CHECK SUN
                  *E60 046    *                       BCPL       RESIDENT CHANNEL CORE LOAD ROUTINE
                  *E60 050    *                       BDIA       BASIC DIAGNOSTIC...RESIDENT CPU DIAGNOSTIC.
                  *E60 051    *                       BMCK       MACHINE CK TRAP ROUTINE
                  *E60 052    *                       BPSW       PSW STORE AND LOADING
                  *E60 053    *                       BSTP       INSTRUCTION STEP TYPE OUT-INSTRUCTION ADDRESS
                  *E60 054    *                       BSWI       EXCEPTIONAL CONDITION ROUTINE
                  *E60 055    *                       BSYS       SYSTEM RESET ROUTINE
                  *E60 056    *                       BWRP       WRAP TRAP ROUTINES
                  *E60 057    *                       CBIN       BINARY ARITH.-ADD-SUB-COMPARE
                  *E60 058    *                       CBRC       BRANCH INSTRUCTIONS
                  *E60 059    *                       CCOM       CPU COMMON ROUTINES
                  *E60 060    *                       CDMD             DECIMAL MULT DIVD
                  *E60 061    *                       CDVD       BINARY DIVIDE ROUTINE
                  *E60 063    *                       CFAD       FLOATING POINT ADD,SUBTRACT,COMPARE
                  *E60 064    *                       CFCO       FLOATING POINT COMMON ROUTINES
                  *E60 066    *                       CFCY       FLOATING POINT INSTRUCTION CYCLES
                  *E60 068    *                       CFDV       FLOATING POINT DEVIDE
                  *E60 069    *                       CFHA       FLOATING POINT HALVE
                  *E60 070    *                       CFLS       FLOATING POINT LOADS AND STORE
                  *E60 071    *                       CFMD       FLOATING POINT MULT DIV PREPARATION
                  *E60 072    *                       CFMU       FLOATING POINT MULTIPLY
                  *E60 074    *                       CICY       INSTRUCTION CYCLES
                  *E60 075    *                       CLOG       LOGICAL OPERATIONS
                  *E60 076    *                       CLST       LOAD AND STORE OPERATIONS
                  *E60 077    *                       CMLT       BINARY MULTIPLY ROUTINE
                  *E60 078    *                       CMPU       PACK UNPACK AND MOVE WITH OFFSET
                  *E60 079    *                       CNVR       CONVERT TO DECIMAL AND BINARY
                  *E60 080    *                       CSFT       SHIFT ROUTINES
                  *E60 081    *                       CSAS       DECIMAL ADD SUB COMPARE ZERO AND ADD
                  *E60 082    *                       CSTS       STATUS SWITCHING INSTRUCTIONS
                  *E60 083    *                       CTRT       TRANSLATE TRANS TEST EDIT EDIT AND MARK
                  *E60 084    *                       DCHN       CHANNEL 0 INTERRUPT ROUTINE----H.BERKEBILE
                  *E60 085    *                       DCLA       I/O INSTRUCTIONS
                  *E60 086    *                       DCLB       CCW FETCH AND CHECK
                  *E60 088    *                       DCLC       CHANNEL INITIAL SELECTION
                  *E60 090    *                       DCLD       CHANNEL DATA LOOP
                  *E60 091    *                       DCLE       CHANNEL LOW PRIORITY TRAP-STATUS ROUTINE
                  *E60 093    *                       DCLH       CHANNEL HALT I/O
                  *E60 095    *                       DCLL       INITIAL PROGRAM LOAD
                  *E60 096    *                       DCLR       CHANNEL ERROR ROUTINE
                  *E60 098    *                       DCLT       CHANNEL TEST I/O AND I/O INTERRUPT
                  *E60 100    *                       DCOM         COMMON I/O ROUTINES
                  *E60 102    *                       DMCS       1403 MULTIPLE CHARACTER SET ROUTINE
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                  *E60 104    *                   DPTC       1403 BUFFER SCAN ROUTINE
                  *E60 105    *                   DPTQ       1403 REQUEST AND INTERRUPT BUFFER ROUTINE
                  *E60 107    *                   DPTR       1403 READ AND WRITE-CHANNEL END AND CARR.CTRL
                  *E60 108    *                   DPTS       1403 START I/O-NO ACT-NO SEC+SENSE COMM
                  *E60 109    *                   DPTT       1403 TEST I/O,HALT I/O,START I/O-SEC. ON
                  *E60 112    *                   DUMP       RESIDENT CORE DUMP ON NATIVE 1403
                  *E60 114    *                   DYPE       1052 TYPE ROUTINES
                  *E60 116    *                   ECOL       PUNCH COLUMN  BINARY AND DATA CHAIN
                  *E60 117    *                   EPCH       PUNCH SIO CMD DECODE & CHAINING
                  *E60 118    *                   EPXF       PUNCH TRANSFER-EBCDIC TO ROW FORM -& CHAINING
                  *E60 119    *                   ERCX       XFER FROM COLUMN FORM BUFFER-RO-TO CCW ADDRESS
                  *E60 120    *                   ERDR       READER START I/O COMMAND DECODE AND CHAINING
                  *E60 122    *                   ERRQ       READER REQUEST MODE AT 10
                  *E60 123    *                   ETRP       NATIVE PUNCH TRAP ROUTINE(PUNCH CHECK ANALYSIS)
                  *E60 124    *                   ETTR       NATIVE READER TRAP ROUTINE(READER CHECK ANALYSIS)
                  *E60 126    *                   EXFR       READER XFER OF ROW IMAGE TO COLUMN BINARY FORM
                  *E60 135    *                   FILE       2311 FEAT., ENTRY FOR SIO, TIO , HIO
                  *E60 137    *                   FILT       2311 FEAT., TRAP ROUTINE
                  *E60 139    *                   FILX       2311 FEAT., OP VERIVICATION & EXECUTION
                  *E60 140    *                   FINT       2311 FEAT., INTERRUPT ROUTINE
                  *E60 257    *                   OVLY       CUSTOMER ROUTINE TO CORRECT BAD MODULE ZERO WORDS
```

AUX STORAGE -- MODULE 0



A----1403 UNIT ADDRESS
B----READER UNIT ADDRESS
C----PUNCH UNIT ADDRESS
D----1052 UNIT ADDRESS
E----H1 SAVE

G----1052 SENSE
N----2311 ADDRESS

H----MULT/DVD TABLE (X1) AND ALT/DIS REGS BACKUP
J----MULT/DVD TABLE (X4) AND ALT/DIS BACKUP + FLPT SAVE
K----MULT/DVD TABLE (X16)
L----MULT/DVD TABLE (X64),FLPT SAVE
M----NATIVE KEY KKKK0000
P----COMMU. Q-EXIT POINTER
R----BURST CHANNEL BUFFERED DEVICE ADDRESSES.

K-ADDRESSABLE AREA

K0-88- CHANNEL 1 INTERRUPT BUFFER

K1-8A- STANDARD INTERFACE, NEXT CCW ADDRESS

K2-8C- CHANNEL UNIT ADDRESS BUFFER

K3-8E- STATUS/ACTIVE BYTE FOR 2311 OR CHANNEL 1

K4-98- 2311 NEXT CCW ADDRESS

K5-9A- 2311 SENSE OR PREVIOUS OP AND MASK

K6-9C- 2311 SENSE OR FILE ADDRESS

K7-9E- DIAGNOSTICS, ALTER/DISPLAY BAL BACKUP

K8-A8- SYSTEM MASK -A9- CPU KEY AND AMWP

K9-AA- EXECUTE INSTRUCTION COUNTER ---I REGISTER BACKUP

KA-AC- U REGISTER BACKUP Q FLTPT SAVE DURING INSTRUCTIONS

KB-AE- G REGISTER BACKUP

KC-B8- P REGISTER BACKUP

KD-BA- CHANNEL 0 INTERRUPT BUFFER

KE-BC- ADDRESS OF STRAIGHT MULT/DVD , FLPT SAVE 2540 REGS BACKUP

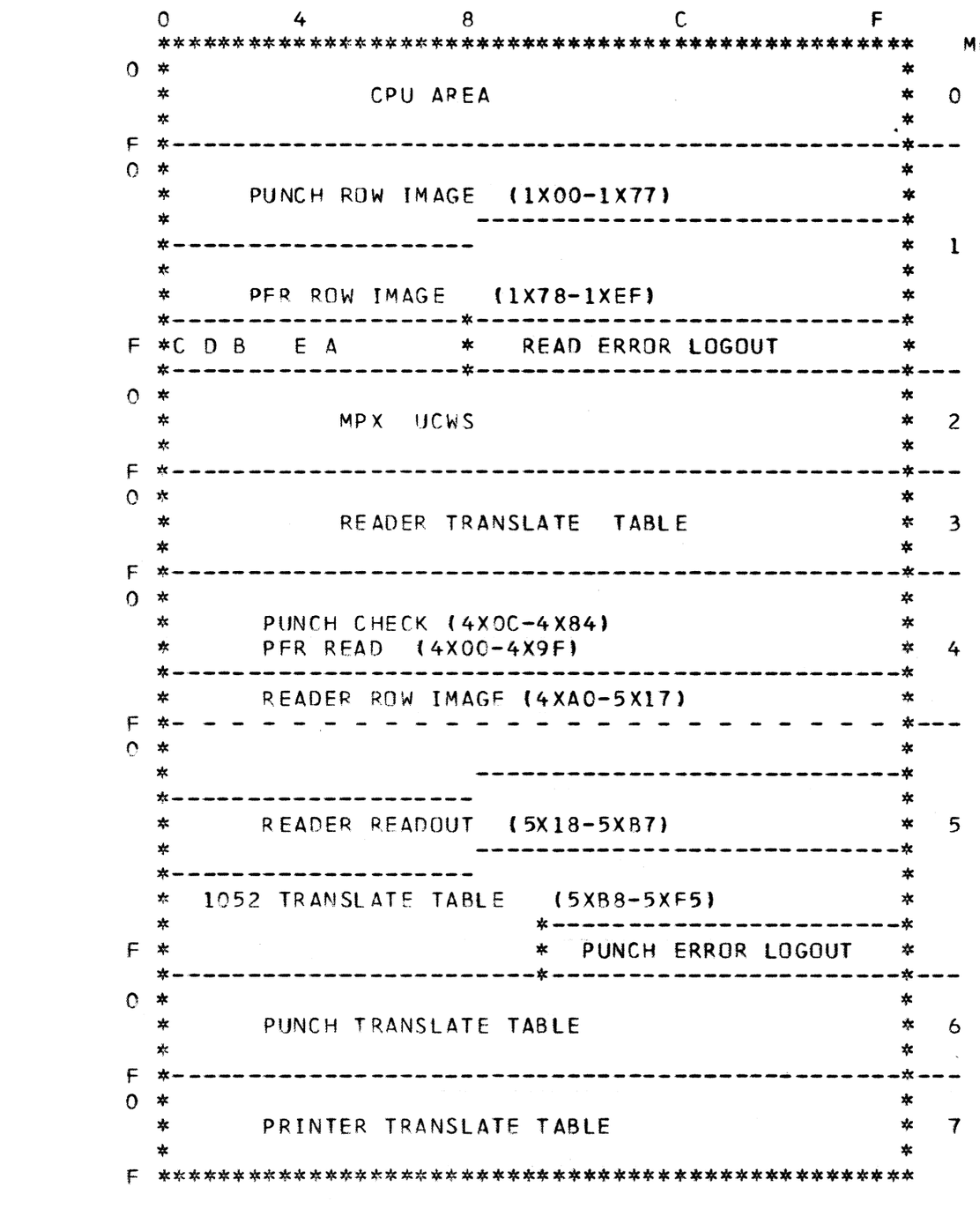
KF-BE- ADDRESS OF SKEWED MULT/DVD , FLTPT SAVE OR CAW KEY. 2540 REGS BACKUP

********** STANDARD DEVICE ADDRESSES **********

NATIVE	LOC	BURST CHANNEL	LOC
1403-------0E	B4	1443 OR 1445-------0B	84
2540 RDR---0C	B5	2540 RDR-----------0C	85
2540 PCH---0D	B6	2540 PCH-----------0D	86
1052-------1F	B7	1403---------------0E	87
2311-------9X	F5	1404 OR 2ND 1403---0F	94
		2520---------------15	95

THE TEXT PRECEEDING THE -BCPL- ROUTINE CONTAINS INFORMATION ON PUNCHING CARDS TO RECONFIGURATE THE SYSTEM TO OTHER THAN THE STANDARD ASSIGNMENTS

AUXILIARY STORAGE ASSIGNMENTS



0	MODULE 0	DEFINED IN AAAB ROUTINE
1	MODULE 1	C AND D NOTED IN TABLE ARE DEFINED IN COMMENTS PRECEEDING THE ERDR ROUTINE. C AT 10F0 D AT 10F1 B, E, AND A NOTED IN TABLE ARE DEFINED IN TEXT PRECEEDING THE EPCH ROUTINE. B AT 10F2, E AT 10F4, A AT 10F5
2	MODULE 2	DEFINED IN FE MAINTENANCE MANUAL
3	MODULE 3	DEFINED AT END OF THE ERCX ROUTINE.
4	MODULE 4	WORKING AREA
5	MODULE 5	1052 TRANSLATE TABLE DEFINED AT END OF THE DYPE ROUTINE.
6	MODULE 6	DEFINED AT END OF THE EPXF ROUTINE. PUNCH TRANSLATE FOR 24K IN MODULE 8.
7	MODULE 7	DEFINED AT END OF THE DPTR ROUTINE FOR NO MCS FEATURE. PRINTER TRANSLATE TABLE FOR 24K IN MODULE 9. PCCL AND BLOCK DATA CHK IN 7X00 OR 9X00.

* AAAC * 360 MODE LOCAL STORAGE ZONES AND ASSIGNMENTS * AAAC *

AS/BS FIELD	NAME	ZONE 0 CPU	ZONE 1 2311	ZONE 4 BACK -UPS	ZONE 5 COMMUNICATIONS	ZONE 6 2540	ZONE 7 CHANNEL
0000	U0	1ST OPERAND STORAGE ADDR.		LEVEL 1 BACK-UP	LCW ADDR HIGH LCW ADDR LOW	READER ROW FORM BUFFER ADDRESS	COUNT
0001	U1						
0010	V0	2ND OPERAND STORAGE ADDR		LEVEL 2 BACK-UP	CHAR POINT HI CHAR POINT LOW	PUNCH IMAGE BUFFER ADDRESS	DATA ADDRESS
0011	V1						
0100	G0	OP CODE		LEVEL 3 BACK-UP	LCW LCW+1	PCH TRAP COUNT WORKING REG	CHANNEL STATUS OP FLAG BYTE
0101	G1	2ND BYTE INST.					
0110	D0	DATA		MACHINE CHECK BACK-UP	LCW+2 LCW+3	INDICATORS RDR TRAP COUNT	CHANNEL IDENT. NOTE 1
0111	D1	HIGH 8 OF 24 BIT 2ND OPERAND ADDR					
ZONE 4 THRU 7							
1000	I0	INST CTR HIGH	CCW COUNT HIGH	CPU BAL			
1001	I1	INST CTR LOW	CCW COUNT LOW	BACK-UP			
1010	T0		DATA ADDR HIGH	SPARE			
1011	T1	GP REG ADDRESS	DATA ADDR LOW	SPARE			
1100	P0	CONDITION CODE	GEN. PURPOSE	LEVEL 1			
1101	P1	PROG MASK, AMWP	GEN. PURPOSE	WORKING AREA			
1110	H0	DATA	CCW ADDR HIGH	LEVEL 1			
1111	H1	HIGH 8 OF 24 BIT INST CTR OR 1ST OPERAND ADDR.	CCW ADDR LOW	WORKING AREA			

... THIS AREA OF ZONE 4 IS COMMON TO ZONES 5,6, AND 7.

NOTE 1-- FOR BURST CHANNEL, THIS REGISTER CONTAINS THE UNIT ADDRESS.

FOR BYTE CHANNEL, THIS REGISTER CONTAINS THE UCW ADDRESS.

EXTERNAL MNEMONIC	DEFINITION		DISPLAYABLE FACILITY	SWITCH C	SWITCH D	ACCESSED BY SET/RESET WORD ONLY
BA	BRANCH CONDITIONS	ALL MODES EXT TO CPU	YES	---	7	
BB	SOFT STOP BRANCH CONDITIONS	CPU MODE EXT TO CPU	YES	CPU	F	
BC	EXTERNAL FACILITY	ALL MODES	---	---	---	YES
CADR	COMM ADAPTER DIAGNOSTIC REG	COMM MODE CPU TO EXT	---	---	---	
CCTRL	START-STOP CONTROL	COMM MODE	---	---	---	YES
CHI	COUNTER 1 HIGH IN- DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	8	
CLI	COUNTER 1 LOW IN - DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	9	
CPF	READ DIRECT ENABLE	CPU MODE	---	---	---	YES
CSETF	START STOP/SYNC	COMM MODE	---	---	---	YES
DAIN	DATA IN	COMM MODE EXT TO CPU	YES	COMM	A	
DAOUT	DATA OUT	COMM MODE CPU TO EXT	---	---	---	
DASI	DISK ATTACHMENT STATUS IN	2311 MODE EXT TO CPU	YES	2311	5	
DIAB	DIAGNOSTIC REG	2311 MODE	---	---	---	YES
DIAC	DIAGNOSTIC REG	2311 MODE	---	---	---	YES
DILIN	DIAL IN	COMM MODE EXT TO CPU	YES	COMM	E	
DILOUT	DIAL OUT	COMM MODE CPU TO EXT	---	---	---	
DR	DIAGNOSTIC REGISTER	ALL MODES	YES	CPU	C	NOTE 1
DS	DISK STATUS	2311 MODE EXT TO CPU	YES	2311	E	
DYN	DYNAMIC CONDITION REGISTER	ALL MODES	YES	---	4	
FBI	FILE BUS IN	2311 MODE EXT TO CPU	YES	2311	1	
FBO	FILE BUS OUT	2311 MODE CPU TO EXT	---	---	---	
FEBO	FILE 1400 EMULATOR BUS OUT	2311 MODE CPU TO EXT	---	---	---	
FFI	FILE FLAGS IN	2311 MODE EXT TO CPU	YES	2311	D	
FFO	FILE FLAGS OUT	2311 MODE CPU TO EXT	---	---	---	
FGA	FILE GATED ATTENTION	2311 MODE EXT TO CPU	YES	2311	C	
FIA	FILE INFORMATION	2311 MODE	---	---	---	YES
FIB	FILE INFORMATION	2311 MODE	---	---	---	YES
FIC	FILE INFORMATION	2311 MODE	---	---	---	YES
FOB	FILE OUT BUS-DIAGNOSTIC	2311 MODE EXT TO CPU	YES	2311	3	
FOP	FILE OP REGISTER	2311 MODE	YES	2311	F	
GA	CHANNEL SIGNALS	CHAN MODE	---	---	---	YES
GB	CHANNEL SIGNALS	CHAN MODE	---	---	---	YES
GB/IN	CHANNEL BUS IN	CHAN MODE EXT TO CPU	YES	CHNL	F	
GB/OUT	CHANNEL BUS OUT	CHAN MODE CPU TO EXT	---	---	---	
GC	CHANNEL SIGNALS	CHAN MODE	---	---		YES
GD	CHANNEL DIAGNOSTIC REGISTER	CHAN MODE EXT TO CPU	YES	CHNL	E	
GS	CHANNEL BRANCH CONDITIONS	CHAN MODE EXT TO CPU	YES	CHNL	C	
GSTAT	GENERAL STATUS	COMM MODE EXT TO CPU	YES	COMM	F	
GT	CHANNEL BRANCH CONDITIONS	CHAN MODE EXT TO CPU	YES	CHNL	D	
JA	DIRECT CONTROL-2 TIMING	CPU MODE CPU TO EXT	---	---	---	
JI	DIRECT CONTROL IN	CPU MODE EXT TO CPU	YES	CPU	8	
JO	DIRECT CONTROL-1	CPU MODE CPU TO EXT	---	---	---	
LACON	LINE ADAPTER CONDITIONS	COMM MODE EXT TO CPU	YES	COMM	C	
LADR	LINE ADAPTER DIAG REGISTER	COMM MODE CPU TO EXT	---	---	---	
LAIN	LINE ADDRESS IN	COMM MODE EXT TO CPU	YES	COMM	B	
LAOUT	LINE ADDRESS OUT	COMM MODE CPU TO EXT	---	---	---	
LASTAT	LINE ADAPTER STATUS	COMM MODE EXT TO CPU	YES	COMM	D	

EXTERNAL MNEMONIC	DEFINITION		DISPLAYABLE FACILITY	SWITCH C	SWITCH D	ACCESSED BY SET/RESET WORD ONLY	
MC	ERROR REGISTER	CPU MODE EXT TO CPU	YES	CPU	E		
MMSK	MICROPROGRAM MASK REGISTER	ALL MODES	YES (0-7)	---	6	NOTE 2	
MODE	MODE REG (LS AND EXT ADDR CTRL)	ALL MODES				NOTE 3	
MS	MODULE SELECT REGISTER	2311 MODE CPU TO EXT	---	---	---		
MW	MACH CHK, WAIT STATE LATCHES	CPU MODE CPU TO EXT	---	---	---		
P	PUNCH SIGNALS	2540 MODE	---	---	---	YES	
PARCK	COMMUNICATIONS PARITY CHECK	COMM MODE CPU TO EXT	---	---			
PCCL	PRINT CHAR COUNTER LENGTH	1403 MODE CPU TO EXT	---	---	---		
PO	PUNCH DATA OUT (READER)	2540 MODE CPU TO EXT	---	---	---		
PR	1403 PLBAR DATA OUT	1403 MODE CPU TO EXT	---	---	---		
PRA	PRINTER SIGNALS	1403 MODE	---	---	---	YES	
PRB	PRINTER SIGNALS	1403 MODE	---	---	---	YES	
PRC	1403 CARRIAGE DATA OUT	1403 MODE CPU TO EXT	---	---	---		
PRD	1403 DIAGNOSTIC CONDITIONS	1403 MODE EXT TO CPU	YES	1403	F		
PRI	1403 PLB DATA IN	1403 MODE EXT TO CPU	YES	1403	B		
PRO	1403 PLB DATA OUT	1403 MODE CPU TO EXT	---	---	---		
PRS	SENSE/STATUS CONDITIONS	1403 MODE EXT TO CPU	YES	1403	E		
PRT	PLBAR DATA IN	1403 MODE EXT TO CPU	YES	1403	A		
PS	PUNCH BRANCH CONDITIONS	2540 MODE EXT TO CPU	YES	2540	F		
R	READER SIGNALS	2540 MODE	---	---	---	YES	
RP	2540 SIGNALS	2540 MODE	---	---	---	YES	
RP1	READER/PUNCH DATA IN 1	2540 MODE EXT TO CPU	YES	2540	B		
RP2	READER/PUNCH DATA IN 2	2540 MODE EXT TO CPU	YES	2540	A		
RPD	2540 SIGNALS DIAG	2540 MODE	---	---	---	YES	
RPD1	DIAGNOSTIC R/P CONDITIONS 1	2540 MODE EXT TO CPU	YES	2540	8		
RPD2	DIAGNOSTIC R/P CONDITIONS 2	2540 MODE EXT TO CPU	YES	2540	9		
RPS	READER/PUNCH BR CONDITIONS	2540 MODE EXT TO CPU	YES	2540	E		
RS	READER BRANCH CONDITIONS	2540 MODE EXT TO CPU	YES	2540	D		
S	STATUS REGISTER	ALL MODES	YES	NOT 2311	5	NOTE 4	
SDI	SERIALIZER/DESERIALIZER IN DIAG	2311 MODE EXT TO CPU	YES	2311	B		
SM	SYSTEM MASK	CPU MODE CPU TO EXT	---	---	---		
STPO	STORAGE PROTECT KEY	ALL MODES	YES	SEE NOTE 5			
STP1	STORAGE PROTECT STACK	ALL MODES(NOT 2311)	YES	SEE NOTE 5			
SWAB	CONSOLE ADDRESS SWITCHES A-B	CPU MODE EXT TO CPU	---	---	---		
SWCD	CONSOLE ADDRESS SWITCHES C-D	CPU MODE EXT TO CPU	---	---	---		
TA	1052 SIGNALS	1052 MODE	---	---	---	YES	
TC	TERMINATING CONDITIONS	2311 MODE EXT TO CPU	YES	2311	A		
TD	PRKB DIAGNOSTIC REGISTER	1052 MODE EXT TO CPU	YES	PRKB	D		
TE	PRKB DATA OUT	1052 MODE CPU TO EXT	---	---	---		
TGRI	TAG REGISTER IN	2311 MODE EXT TO CPU	YES	2311	0		
TGRO	TAG REGISTER OUT	2311 MODE CPU TO EXT	---	---	---		
TI	1052 DATA IN	1052 MODE EXT TO CPU	YES	PRKB	A		
TIM	TIMER COUNT	CPU MODE EXT TO CPU	YES	CPU	A		
TR	1052 TILT/ROTATE REGISTER	1052 MODE EXT TO CPU	YES	PRKB	B		
TT	PRKB BRANCH CONDITIONS	1052 MODE EXT TO CPU	YES	PRKB	E		
TU	PRKB BRANCH CONDITIONS	1052 MODE EXT TO CPU	YES	PRKB	F		
XINT	EXTERNAL INTERRUPT	CPU MODE EXT TO CPU	YES	CPU	9		

NOTE 1 THE DR REGISTER IS SET BY THE SET/RESET WORD. EXECUTION OF A BRANCH WORD OR A RETURN WORD, WITH DR BIT 7 ON, SETS THE DIAGNOSTIC BRANCH LATCH.

NOTE 2 THE MMSK REGISTER 0-9 IS SET OR RESET BY THE SET/RESET WORD.
FOR OTHER WORD ACCESSES AND DISPLAY, ONLY BITS 0-7 CAN BE ADDRESSED.

NOTE 3 THE MODE REGISTER IS SET BY THE SLT/RESET WORD. THE ACTUAL BIT STRUCTURE OF THE MODE REGISTER IS NOT ALWAYS WHAT IS DISPLAYED IN THE CONSOLE INDICATORS LABELED MODE/ZONE REG. THE MODE BIT DECODE IS DISPLAYED IN BITS 2, 3,4, AND THE LS ZONE BIT DECODE IS DISPLAYED IN BITS 5,6,AND 7. THESE COULD BE THE FORCED DECODES CAUSED BY AN MMSK BIT BEING SET.
** REFER TO DIAGRAMS 4-14 AND 4-32 IN THE -FEMDM- FOR DISPLAY DETAILS.

NOTE 4 THE S-REGISTER IS SET OR RESET BY THE SET/RESET WORD. IT CAN ALSO BE ACCESSED BY A BRANCH WORD.
BITS 0-6 CAN BE DISPLAYED DIRECTLY BUT BIT 7 OF THE DISPLAY INDICATES THE STATUS OF THE S7 BRANCH CONDITION LINE. TO DISPLAY THE ACTUAL CONDITION OF THE S7 LATCH, THE BA FACILITY MUST BE DISPLAYED AND BIT 0 WILL INDICATE THE S7 LATCH STATUS.

NOTE 5 TO DISPLAY STP1 FOR MPX OPERATION--
1. SET CONSOLE SWITCHES A,B,C, AND D TO THE AUXILIARY STORAGE ADDRESS OF THE MPX UCW ASSOCIATED WITH THE I/O DEVICE.
2. SET MODE SWITCH TO AS DATA.
3. PRESS AND RELEASE THE DISPLAY KEY. (M-REG SET TO ADDRESS IN SW,S ABCD).
4. SET MODE SWITCH TO ALU/EXT.
5. SET CONSOLE SWITCHES C,D TO 03.
6. PRESS AND HOLD THE DISPLAY KEY. THE ASSOCIATED I/O PROTECT KEY IS DISPLAYED IN THE BYTE 1 INDICATORS.

TO DISPLAY STP1 FOR SELECTOR CHANNEL KEY
1. SET CONSOLE SWITCHES A,B,C, AND D TO AUX ADDRESS XX88. THE REST OF THE DISPLAY PROCEEDURE IS THE SAME AS THE MPX STP1.

TO DISPLAY STP1 FOR CPU STORAGE KEY
1. SET CONSOLE SWITCHES A,B,C, AND D TO A MAIN STORAGE ADDRESS ASSOCIATED WITH A PARTICULAR STORAGE KEY.
2. SET MODE SWITCH TO MS DATA. THE REST OF THE DISPLAY PROCEEDURE IS THE SAME AS THE MPX STP1.

TO DISPLAY THE FILE STORAGE KEY (WHICH IS NOT IN THE STP1 STACK) DO THE FOLLOWING--
1. SET SWITCHES C,D TO 12.
2. SET MODE SWITCH TO ALU/EXT.
3. PRESS AND HOLD THE DISPLAY KEY. THE FILE KEY IS DISPLAYED IN THE BYTE 1 INDICATORS.

TO DISPLAY STP0
1. SET SWITCH C TO MODE DESIRED--
   C=0=CPU MODE (Q REG)
   C=1=2311 MODE (FQ REG)
   C=5=COMM MODE (HQ REG)
   C=7=CHAN MODE (GQ REG)
2. SET SWITCH D TO 2.
3. SET MODE SWITCH TO ALU/EXT.
4. PRESS AND HOLD THE DISPLAY KEY.

WORD TYPE	K HIGH	SET/RST SOURCE FIELD	B SOURCE OR K LOW FIELD	K HIGH FIELD
0 1	2	3 4 5 6 7	8 9 10 11	12 13 14 15
0 0	0=RST 1=SET / K HIGH 0 BIT	0000= S REGISTER ALL MODES 0010= MMSK REGISTER ALL MODES 0100= MODE REGISTER ALL MODES 0110= BC FACILITY ALL MODES 1000= DR REGISTER ALL MODES 1001= RPD FACILITY 2540 MODE 0001= DIAB FACILITY 2311 MODE 1011= R FACILITY 2540 MODE 1011= GA FACILITY CHNL MODE 0011= DIAC FACILITY 2311 MODE 1101= PRA FACILITY 1403 MODE 1101= FIA FACILITY 2311 MODE 1101= RP FACILITY 2540 MODE 1101= CSETF FACILITY COMM MODE 1101= GB FACILITY CHNL MODE 1110= FIB FACILITY 2311 MODE 1111= CPF FACILITY CPU MODE 1111= PRB FACILITY 1403 MODE 1111= TA FACILITY 1052 MODE 1111= FIC FACILITY 2311 MODE 1111= P FACILITY 2540 MODE 1111= CCTRL FACILITY COMM MODE 1111= GC FACILITY CHNL MODE	WHEN MMSK ADDRESSED FOR A LINK OR RETURN FUNCTION, THE B SOURCE DECODES ARE FOR LOCAL STORAGE 0000= U REGISTER 0010= V REGISTER 0100= G REGISTER 0110= D REGISTER 1000= I REGISTER 1010= T REGISTER 1100= P REGISTER 1110= H REGISTER WHEN USED AS THE K FIELD BIT 8 = K LOW BIT 0 BIT 9 = K LOW BIT 1 BIT 10= K LOW BIT 2 BIT 11= K LOW BIT 3	1 BIT 2 BIT 3 BIT / 0

WHEN THE MMSK REGISTER IS ADDRESSED BY THE SET/RST SOURCE FIELD, CONTROL WORD BIT 11=0 INDICATES A LINK OR RETURN FUNCTION, BIT 11=1 INDICATES NO LINK OR RETURN FUNCTION.

BIT SIGNIFICANCE CHART FOR THE SET/RST WORD (WORD TYPE 0)

WORD TYPE	FUNCTION DECODE	A-SOURCE FIELD	K FIELD	FUNCTION DECODE	WORD TYPE
0 1	2 3	4 5 6 7	8 9 10 11	12 13 14	15
0 0	0 0	0000 = U0	0000	000 - Z=A + KL	1
	0 0	0001 = U1	0001	001 - Z=A,A-,KL	
	0 0	0010 = V0	0010	010 - Z=A,A-,KH	
	0 0	0011 = V1	0011	011 - Z=A,A-,KK	
	0 0	0100 = G0	0100	100 - Z=A + KH	
	0 0	0101 = G1	0101	101 - Z=A,OE,KL	
	0 0	0110 = D0	0110	110 - Z=A,OE,KH	
	0 0	0111 = D1	0111	111 - Z=A,OE,KK	
	0 1	1000 = I0	1000	000	
	0 1	1001 = I1	1001	001 - A=A,A-,KL	
	0 1	1010 = T0	1010	010 - A=A,A-,KH	
	0 1	1011 = T1	1011	011 - A=A,A-,KK	
	0 1	1100 = P0	1100	100	
	0 1	1101 = P1	1101	101 - A=A,OE,KL	
	0 1	1110 = H0	1110	110 - A=A,OE,KH	
	0 1	1111 = H1	1111	111 - A=A,OE,KK	
	1 0			000 - A=0 - KL	
	1 0			001 - A=0,OR,KL	
	1 0			010 - A=0,OR,KH	
	1 0	THE FUNCTION DECODE IS A		011 - A=0,OR,KK	
	1 0	COMBINATION OF CONTROL WORD		100	
	1 0	BITS 2, 3, 12, 13, AND 14.		101 - A=A + KL	
	1 0			110 - A=A + KH	
	1 1	THE -A- SYMBOL IN THE		000	
	1 1	FORMULAS UNDER COLUMNS 12, 13,		001 - A=A,OR,KL	
	1 1	AND 14 CAN BE ANY OF THE LOCAL		010 - A=A,OR,KH	
	1 1	STORAGE BYTE SOURCES LISTED		011 - A=A,OR,KK	
	1 1	UNDER THE A-SOURCE FIELD.		100 - A=0 - KH	
	1 1			101 - A=A - KL	
	1 1			110 - A=A - KH	
	1 1			111 - A=A - KK	

BIT SIGNIFICANCE CHART FOR THE ARITH CONSTANT WORD (WORD TYPE 1)

WORD TYPE	STORAGE CONTROL	DATA REGISTER ADDRESS	ADDRESS REGISTER OR K FIELD	MODIFIER CONTROL	WORD TYPE
0 1	2 3	4 5 6 7	8 9 10 11	12 13 14	15
0 1	00=READ CONTROL 01=READ AUX OR PROGRAM 10=STORE CNTL 11=STORE AUX OR PROGRAM BIT 11 DETERMINES IF THE AUXILIARY STORAGE AREA OR THE PROGRAM STORAGE AREA IS ACCESSED FOR BIT 2,3 DECODES OF 01 AND 11. THIS IS ONLY TRUE WHEN INDIRECTLY ADDRESSING. WHEN NO ACESS TO STORAGE IS DESIGNATED BY THE MODIFIER CONTROL FIELD, BITS 2 AND 3 INDICATE THE UPDATE VALUE. BITS 2,3 — VALUE 00 — 0 01 — + OR - 1 10 — + OR - 2 11 — + OR - 2	THIS FIELD MAY ADDRESS LOCAL STORAGE FOR A BYTE OR HALFWORD. BYTE SELECTION OF LOCAL STORAGE IS LIMITED TO THE ODD ADDRESSES. HALFWORD SELECTION IS LIMITED TO EVEN ADDRESSES. EXTERNAL FACILITY ADDRESSING IS ALWAYS DONE IN BYTE MODE, THEREFOR ONLY THE EXTERNALS WITH ODD ADDRESSES CAN BE ACCESSED. LOCAL STORAGE DECODES 0000 - U0 0001 - U1 0010 - V0 0011 - V1 0100 - G0 0101 - G1 0110 - D0 0111 - D1 1000 - I0 1001 - I1 1010 - T0 1011 - T1 1100 - P0 1101 - P1 1110 - H0 1111 - H1 ALL EVEN ADDRESSES CAN BE USED AS ADDRESS REGISTERS IN THE FIELD DESIGNATED BY CONTROL WORD BITS 8,9,10,AND 11.	FOR INDIRECT ADDRESSING THE ADDRESS REGISTER SELECTED BY THIS FIELD IS A HALFWORD LOCAL STORAGE REGISTER BIT 11 DOES NOT ENTER INTO THE ADDRESSING OF THE LOCAL STORAGE REGISTER, BUT DOES INDICATE THE FOLLOWING BIT 11 — BITS 2,3 — MEANING 0 — 01 — READ AUXILIARY 1 — 01 — READ PROGRAM 0 — 11 — STORE AUXILIARY 1 — 11 — STORE PROGRAM FOR DIRECT ADDRESSING THIS FIELD CONTAINS BIT CODES THAT FORCE THE ADDRESS REGISTER M1 TO SPECIFIC VALUES THESE VALUES ARE- BITS 8,9,10,11 — FORCED M1 VALUES 0000 — 88 0001 — 8A 0010 — 8C 0011 — 8E 0100 — 98 0101 — 9A 0110 — 9C 0111 — 9E 1000 — A8 1001 — AA 1010 — AC 1011 — AE 1100 — B8 1101 — BA 1110 — BC 1111 — BE	000 - NO UPDATE 001 - DIRECT ADDRESSING 010 - NO ACCESS, PLUS UPDATE. 011 - NO ACCESS, MINUS UPDATE. 100 - LOCAL STORAGE DATE REG, ACCESS, PLUS UPDATE. 101 - LOCAL STORAGE DATE REG, ACCESS, MINUS UPDATE. 110 - EXTERNAL DATA REG, ACCESS, PLUS UPDATE. 111 - EXTERNAL DATA REG, ACCESS, MINUS UPDATE. NOTE-- WHEN DIRECT ADDRESSING CONTROL STORAGE, M0 IS SET TO X3. WHEN DIRECT ADDRESSING AUXILIARY STORAGE, M0 IS SET TO 00.	0

SIGNIFICANCE CHART FOR THE STORAGE WORD (WORD TYPE 2)

WORD TYPE	FUNCTION DECODE	A SOURCE FIELD	B SOURCE FIELD	FUNCTION DECODE	WORD TYPE
0 1	2 3	4 5 6 7	8 9 10 11	12 13 14	15
0 1	0 0	0000 = U0	0000 = U0	000	1
	0 0	0001 = U1	0001 = U1	001 - B=AXH + BL	
	0 0	0010 = V0	0010 = V0	010 - B=AXL,OR,BH	
	0 0	0011 = V1	0011 = V1	011	
	0 0	0100 = G0	0100 = G0	100 - B=A + B	
	0 0	0101 = G1	0101 = G1	101 - B=AH + BL	
	0 0	0110 = D0	0110 = D0	110 - B=AL + BH	
	0 0	0111 = D1	0111 = D1	111 - EXT=B	
	0 1	1000 = I0	1000 = I0	000 - B=AX	
	0 1	1001 = I1	1001 = I1	001 - B=AXH	
	0 1	1010 = T0	1010 = T0	010 - B=AXL	
	0 1	1011 = T1	1011 = T1	011 - B=0 (STOP)	
	0 1	1100 = P0	1100 = P0	100 - B=A	
	0 1	1101 = P1	1101 = P1	101 - B=AH	
	0 1	1110 = H0	1110 = H0	110 - B=AL	
	0 1	1111 = H1	1111 = H1	111 - B=EXT	
	1 0	THE FUNCTION DECODE IS A COMBINATION OF CONTROL WORD BITS 2,3,12,13, AND 14.		000 - A=A,OE,B	
	1 0			001 - A=A + B	
	1 0			010 - A=A,OR,B	
	1 0	IF THE FUNCTION DECODE IS EQUAL TO 00111, THE A-SOURCE FIELD ADDRESSES AN EXTERNAL FACILITY.		011 - A=A,A,B	
	1 0			100 - AC=A + B+1	
	1 0	IF THE FUNCTION DECODE IS EQUAL TO 01111, THE A-SOURCE FIELD ADDRESSES AN EXTERNAL FACILITY.		101 - AC=A + B	
	1 0			110 - AC=A + B+C	
	1 0			111 - AC=AL + B+C	
	1 1	THE SYMBOL -A- USED IN THE FUNCTION DECODE FORMULAS, REFERS TO ANY OF THE LOCAL STORAGE SYMBOLS UNDER THE A-SOURCE FIELD.		000 - A=A - B+1	
	1 1	THE SYMBOL -B- USED IN THE FUNCTION DECODE FORMULAS, REFERS TO ANY OF THE LOCAL STORAGE SYMBOLS UNDER THE B-SOURCE FIELD.		001 - A=A - B	
	1 1			010	
	1 1			011	
	1 1	THE SYMBOL -C- REFERS TO THE ADDER CARRY (S3 LATCH)		100 - AC=A - B+C	
	1 1			101 - AC=0 - B+C	
	1 1	THE SYMBOL -%- REFERS TO A BINARY +- OPERATION UNDER S0 CTRL.		110 - A=A % B+C	
	1 1	THE SYMBOL -@- REFERS TO A DECIMAL +- OPERATION UNDER S0 CTRL.		111 - AC=A @ B+C	

BIT SIGNIFICANCE CHART FOR THE MOVE/ARITHMETIC WORD (WORD TYPE 3)

WORD TYPE	REPLACEMENT BITS FOR THE MO-REGISTER	REPLACEMENT BITS FOR THE M1-REGISTER
0 1	2 3 4 5 6 7	8 9 10 11 12 13 14	15
1 0	BITS 2 THRU 7 OF THIS CONTROL WORD CONTAIN THE VALUES THAT ARE GATED TO BITS 2 THRU 7 OF THE MO-REGISTER WHEN THIS WORD IS EXECUTED. WHEN THIS WORD IS GATED INTO THE CONTROL REGISTER, BIT 5 OF THE CONTROL REGISTER IS FORCED TO A 1. THE REPLACEMENT VALUE FOR THE MO-REGISTER BIT 5 POSITION IS GATED FROM THE STORAGE DATA BUS BIT 5 POSITION.	BITS 8 THRU 14 CONTAIN THE VALUES THAT ARE GATED TO BITS 0 THRU 6 OF THE M1-REGISTER WHEN THIS WORD IS EXECUTED.	0

WHEN THIS WORD IS EXECUTED, THE ADDRESS OF THE NEXT SEQUENTIAL CONTROL WORD IS STORED IN THE I-REGISTER OF LOCAL STORAGE ZONE 4.

BIT SIGNIFICANCE CHAR FOR THE BRANCH UNCONDITIONAL WORD (WORD TYPE 4)

WORD TYPE	BRANCH CONTROL	A SOURCE FIELD		REPLACEMENT BITS FOR M1-REGISTER	REPLACEMENT BITS FOR MO-REGISTER.	WORD TYPE
0 1	2 3	4 5 6 7	8	9 10 11	12 13 14	15
1 0	00= 4 WAY BR. 01= 8 WAY BR 10= 2 WAY BR (A SOURCE NOT ZERO) 11= 16 WAY BR	BIT 5=0 MEANS STRAIGHT A-REG GATING. BIT 5=1 MEANS CROSS A-REG OUTPUT --- WHEN THIS CONTROL WORD IS READ INTO THE CONTROL REGISTER, BIT 5 IS FORCED TO 1. THIS RESTRICTS THE FACILITIES THAT CAN BE ADDRESSED BY THE A-SOURCE FIELD TO THOSE THAT HAVE AT LEAST BIT 5=1 IN THEIR A-SOURCE DECODES. BECAUSE BIT 5 IS FORCED IN THE CONTROL REGISTER, THE TRUE STATUS OF BIT 5 IS TAKEN FROM THE STORAGE DATA BUS OUT, FOR A-REG GATING.	0=LS 1=EXT	BIT 9 CONTAINS THE REPLACEMENT VALUE FOR BIT 1 OF THE M1-REGISTER. BIT 10 CONTAINS THE REPLACEMENT VALUE FOR BIT 2 OF THE M1 REGISTER. BIT 11 CONTAINS THE REPLACEMENT VALUE FOR BIT 0 OF THE M1 REGISTER	BITS 12, 13, AND 14 CONTAIN THE REPLACEMENT VALUES FOR BITS 5, 6, AND 7 OF THE MO REG.	1

EITHER DECODE IN CONTROL STORAGE.	LOCAL STORAGE REGISTER
0000 OR 0100	G0
0001 OR 0101	G1
0010 OR 0110	D0
0011 OR 0111	D1
1000 OR 1100	P0
1001 OR 1101	P1
1010 OR 1110	H0
1011 OR 1111	H1

EXTERNAL FACILITIES HAVE THE SAME ADDRESSING RESTRICTIONS.

BIT SIGNIFICANCE CHART FOR BRANCH ON MASK WORD (WORD TYPE 5)

WORD TYPE	BIT CONTROL	A-SOURCE FIELD		REPLACEMENT BITS FOR THE M1-REGISTER							WORD TYPE
0 1	2 3	4 5 6 7	8	9 10 11 12 13 14							15
1 1	00= BIT 0 OR 4 01= BIT 1 OR 5 10= BIT 2 OR 6 11= BIT 3 OR 7	BIT 5=0 MEANS STRAIGHT A-REG GATING. BIT 5=1 MEANS CROSS A-REG OUTPUT.	0=LS 1=EXT	BITS 9--14 CONTAIN THE REPLACEMENT VALUES FOR BITS 1--6 OF THE M1-REGISTER.							0 TEST FOR 0
		WHEN THIS CONTROL WORD IS READ INTO THE CONTROL REGISTER, BIT 5 IS FORCED TO 1. THIS RESTRICTS THE FACILITIES THAT CAN BE ADDRESSED BY THE A-SOURCE FIELD TO THOSE THAT HAVE AT LEAST BIT 5=1 IN THEIR A-SOURCE DECODES. BECAUSE BIT 5 IS FORCED IN THE CONTROL REGISTER, THE TRUE STATUS OF BIT 5 IS TAKEN FROM THE STORAGE DATA BUS OUT, FOR A-REG GATING.									1 TEST FOR 1

EITHER DECODE IN CONTROL STORAGE	LOCAL STORAGE REGISTER
0000 OR 0100	G0
0001 OR 0101	G1
0010 OR 0110	D0
0011 OR 0111	D1
1000 OR 1100	P0
1001 OR 1101	P1
1010 OR 1110	H0
1011 OR 1111	H1

EXTERNAL FACILITIES HAVE THE SAME ADDRESSING RESTRICTIONS.

BIT SIGNIFICANCE CHART FOR BRANCH ON CONDITION WORD (WORD TYPE 6 OR 7)

MICROPROGRAM TEMPORARY FIX RECORD

MPTF NO. _____ EC LEVEL AFFECTED _____ PICKUP EC _____

RETEST PROCEDURE

PROBLEM DEFIFITION

ADDR	WORD	LABLE	BR LABLE	STATEMENT	COMMENTS	SEQUENCE NO.

ORIGINAL CHECK SUM _____ NEW CHECK SUM _____ NO. OF STOP WORDS REMOVED _____

WORDS OVERLAYED BY PATCH

ROUT.SEQ.*ORIG.WORD	ROUT.SEQ.*ORIG.WORD	ROUT.SEQ.*ORIG.WORD

MICROPROGRAM TEMPORARY FIX RECORD

MPTF NO. _____ EC LEVEL AFFECTED _____ PICKUP EC _____

RETEST PROCEDURE

PROBLEM DEFIFITION

ADDR	WORD	LABLE	BR LABLE	STATEMENT	COMMENTS	SEQUENCE NO.

ORIGINAL CHECK SUM _____ NEW CHECK SUM _____ NO. OF STOP WORDS REMOVED _____

WORDS OVERLAYED BY PATCH

ROUT.SEQ.	ORIG.WORD	ROUT.SEQ.	ORIG.WORD	ROUT.SEQ.	ORIG.WORD

MICROPROGRAM TEMPORARY FIX RECORD

MPTF NO. _____ EC LEVEL AFFECTED _____ PICKUP EC _____

RETEST PROCEDURE

PROBLEM DEFIFITION

ADDR	WORD	LABLE	BR LABLE	STATEMENT	COMMENTS	SEQUENCE NO.

ORIGINAL CHECK SUM _____ NEW CHECK SUM _____ NO. OF STOP WORDS REMOVED _____

WORDS OVERLAYED BY PATCH

ROUT.SEQ.*ORIG.WORD	ROUT.SEQ.*ORIG.WORD	ROUT.SEQ.*ORIG.WORD

ALDP -- CONSOLE PRINTER-KEYBOARD ALTER/DISPLAY

ADDITIONAL INFORMATION REFERENCE --
FEMDM DIAGRAMS 5-77 THROUGH 5-79

DESCRIPTION

THIS ROUTINE ALLOWS MANUAL 1052 ALTER OR DISPLAY OF AUXILIARY STORAGE, CONTROL STORAGE, AND PROGRAM STORAGE. ANY NUMBER OF BYTES CAN BE ALTERED. DISPLAYING, HOWEVER, IS ALWAYS IN MULTIPLES OF EIGHT BYTES.

WHEN THE 'PROCEED' LIGHT COMES ON AFTER PRESSING THE PR-KB ALTER DISPLAY BUTTON, TYPE THE FIRST TWO CHARACTERS AS FOLLOWS --

1ST CHARACTER -- A OR D
- A = ALTER
- D = DISPLAY

2ND CHARACTER -- A, C, OR P
- A = AUXILIARY STORAGE
- C = CONTROL STORAGE -- CE MODE ONLY
- P = PROGRAM STORAGE

B OR D, B=DISP 1403 BUFFER(REF. DPTC) D=CORE DUMP(REF. DUMP)

AFTER THE FIRST TWO CHARACTERS, TYPE A FOUR-HEX-DIGIT ADDRESS. THE ROUTINE THEN FORCES A LINE FEED. IF THE OPERATION IS ALTER, THE ROUTINE WAITS FOR THE OPERATOR TO TYPE IN THE DATA TO BE ALTERED. IF THE OPERATION IS DISPLAY, THE ROUTINE STARTS RIGHT AFTER THE LINE FEED, TYPING OUT THE FIRST EIGHT BYTES OF DATA THEN COMING TO A HALT. IF THE DISPLAY OF MORE DATA IS DESIRED, THE OPERATOR CAN PRESS THE 'SPACE' BAR, AND THE NEXT EIGHT SEQUENTIAL BYTES WILL BE DISPLAYED. EITHER OPERATION, ALTER OR DISPLAY, SHOULD BE TERMINATED BY AN EOB.

DESCRIPTION CONTINUED

AT ANY POINT DURING THE PROCESS, THE OPERATOR CAN PRESS THE 'CANCEL' KEY TO TERMINATE THE OPERATION EXCEPT WHILE DATA IS BEING TYPED FOR A DISPLAY OPERATION IN WHICH CASE THE KEYBOARD IS LOCKED. WHEN A 'CANCEL' IS ENCOUNTERED, THE OPERATION RETURNS TO THE BEGINNING OF THE ROUTINE GIVING THE OPERATOR A CHANCE TO TRY AGAIN.

EXCEPT THE FIRST TWO CHARACTERS, ALL THE OTHERS SHOULD BE HEX DIGITS ONLY. BOTH UPPER OR LOWER CASE ARE ALLOWED FOR ALPHABETICAL CHARACTERS. THE FOLLOWING CONDITIONS ARE RECOGNIZED AS ERRORS.

1. A NON-HEX DIGIT TYPED ON THE KEYBOARD.
2. FIRST CHARACTER OTHER THAN A OR C.
3. SECOND CHARACTER OTHER THAN A, C, OR P. THE C IS VALID ONLY IN CE MODE.
4. INVALID STORAGE ADDRESS.

THE ROUTINE SENDS A MESSAGE OF 'INVALID CHAR' OR 'INVALID ADDR', THEN RETURNS TO THE BEGINNING TO GIVE THE OPERATOR ANOTHER TRY.

THIS ROUTINE PROVIDES OFFSETTING WHEN THE ADDRESS IS ODD.

WHEN ONE PASS OF THE OPERATION IS COMPLETED, THE ROUTINE ALWAYS GOES BACK TO THE BEGINNING. THUS, THE OPERATOR CAN DO ADDITIONAL ALTER OR DISPLAY OPERATIONS WITHOUT PRESSING THE ALTER-DISPLAY BUTTON AGAIN, OR CAN PRESS THE 'START' BUTTON ON THE CONSOLE IF HE WISHES TO GO BACK TO THE INTERRUPTED CPU PROCESS.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              ALDP 001    T            1052 DOCUMENTARY ALTER DISPLAY      R. C. HUANG          10/24/67
              ALDP 002    *            THE WORD LABELED AS 'DOIT' IN BCPL ROUTINE WITH FIXED CONTROL STORAGE
              ALDP 003    *                 ADDRESS OF 039C VARIES ACCORDING TO THE TYPE OF ALDP OPERATION
              ALDP 004    *                          AA  -  ALTER AUX.       -   STB H1 AS,U+1   -   7F08
              ALDP 005    *                          AC  -  ALTER CONTROL    -   STB H1 CS,U+1   -   6F08
              ALDP 006    *                          AP  -  ALTER PROGRAM    -   STB H1 U+1      -   7F18
              ALDP 007    *                          DA  -  DISPLAY AUX.     -   RDH H AS,U+2    -   5E08
              ALDP 008    *                          DC  -  DISPLAY CONTROL  -   RDH H CS,U+2    -   4E08
              ALDP 009    *                          DP  -  DISPLAY PROGRAM  -   RDH H U+2       -   5E18
              ALDP 010    *
              ALDP 011    *
              ALDP 012    *            THE REGISTER USAGES IN THIS ROUTINE ARE -
              ALDP 013    *                      U0,U1 - ADDRESS
              ALDP 014    *                      V0    - COUNTER OF NUMBER OF HALF WORDS
              ALDP 015    *                      V1    - OUTPUT DATA
              ALDP 016    *                      G0    - INPUT DATA
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS

            ALDP 017   *                        G1     - COLUMN COUNTER
            ALDP 018   *                        D0,D1  - CONTROL WORD AND FLAGS
            ALDP 019   *                        H0,H1  - DATA
            ALDP 020   *
            ALDP 035   ASEQ    AL07=0C
1A0C  26A3  ALDP 037    START                   D0=0$K0A             SET ALLOW START KEY FLAG
1A0E  2783  ALDP 038                            D1=0$K08             DEFINE ADDR TO BE IN U REG'S
1A10  88BC  ALDP 039                241 LINEFD  BAL                  LINE FEED
1A12  88C4  ALDP 040                246 REDCH1  BAL                  READ 1ST CHARACTER
1A14  3643  ALDP 041                            D0=D0$K04            DISABLE  START KEY
1A16  043D  ALDP 042                            Z=G0<K30             MASK 3X IN PTT CODE
1A18  E0B3  ALDP 043                056 IS3X    BR IF HZ=0           BR IF 1ST CH. IS 3X
1A1A  88BC  ALDP 044    BADCHR      241 LINEFD  BAL                  INVALID CHARACTER MESSAGE
1A1C  24B3  ALDP 045    BADADR                  G0=0$K0B             NEGATIVE G0=F4
1A1E  2577  ALDP 046                            G1=0$K77             SET G=0B77 TO ADDR MESSAGE
1A20  534A  ALDP 047    RDMORE                  RDB V1 AS,G-1        READ OUT MESSAGE FROM AUX. 0
1A22  88CE  ALDP 048                252 TYPCHR  BAL                  TYPE MESSAGE OUT
1A24  D121  ALDP 049                047 RDMORE  BR IF G1 BIT5=1      BR IF NOT TO CHANGE ROW
1A26  D00D  ALDP 050                037 START   BR IF G0 BIT5=1      BR IF MESSAGE COMPLETED
1A28  7543  ALDP 051    CHGROW                  G1=G1-G0             CHANGE ROW BY ADDING F4 OR F0
1A2A  F520  ALDP 052                047 RDMORE  BR IF G1 BIT3=0      BR IF TO PRINT 2ND OR 4TH ROW
1A2C  24F3  ALDP 053                            G0=0$K0F             NEGATIVE G0=F0, & SET DONE FLAG
1A2E  F320  ALDP 054                047 RDMORE  BR IF D1 BIT7=0      BR IF TO PRINT 'CHAR'
1A30  9A28  ALDP 055                051 CHGROW  BR                   GO TO SUBTRACT ONE ROW
1A32  044B  ALDP 056    IS3X                    Z=G0<K04             MASK FOR 'D' - DISPLAY
1A34  F0BD  ALDP 057                061 AORD    BR IF LZ=0           BR IF 1ST CH. IS 'D'
1A36  041B  ALDP 058                            Z=G0<K01             MASK FOR 'A' - ALTER
1A38  F09A  ALDP 059                044 BADCHR  BR IF LZNZ           BR IF 1ST CH. IS NOT 'A'
1A3A  36D9  ALDP 060                            D0=0-KD0             MAKE 2F CONTROL WORD FOR ALTER
1A3C  88C6  ALDP 061     AORD       247 READCH  BAL                  READ 2ND CHARACTER
1A3E  3645  ALDP 062                            D0=DC$K40            D0=4E FOR DSPLY, D0=6F FOR ALTER
            ALDP 063   *
            ALDP 064   *                        D REG'S ARE USED HERE TO SET UP THE VARIABLE CONTROL WORD. THEY
            ALDP 065   *                        ARE ALSO USED IN THIS ROUTINE AS FLAGS. THE USAGE OF EACH FLAG
            ALDP 066   *                        BIT IS AS FOLLOWS -
            ALDP 067   *                         D0 BIT 0 - EXPECTING EOB IN 'READCH' BAL ROUTINE
            ALDP 068   *                                1 - CHAR. READ REQUIRES PACKING IN 'READCH' BAL ROUTINE
            ALDP 069   *                                2 - OPERATION IS ALTER
            ALDP 070   *                                3 - OPERATION IS IN AUX OR PROGRAM STORAGE - NOT CONTROL
            ALDP 071   *                                4 - NOT FROM 'READCH' BAL ROUTINE
            ALDP 072   *                                5 - DISABLE 'START' KEY IN THE SOFT STOP LOOP - BSWI
            ALDP 073   *                                6 - VALID CHARACTER IN 'READCH' BAL ROUTINE
            ALDP 074   *                                7 - NOT USED, 1 FOR ALTER, 0 FOR DISPLAY
            ALDP 075   *                         D1 BIT 0 - ODD ADDRESS
            ALDP 076   *                                1 - LOOPING CONTROL 1
            ALDP 077   *                                2 - LOOPING CONTROL 2
            ALDP 078   *                                3 - OPERATION IS IN PROGRAM STORAGE
            ALDP 079   *                                4 - THE OPERATION IS NOT FROM 'BSTP' ROUTINE FOR
            ALDP 080   *                                      INSTRUCTION ADDRESS TYPE OUT NOR LOG-OUT
            ALDP 081   *                                5 - 16TH CHARACTER
            ALDP 082   *                                6 - LOG OUT FOR MACHINE CK OR CE TRAP
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

            ALDP 083    *                        7 - TO PRINT THE MESSAGE OF 'INVALID ADDR'
            ALDP 084    *
1A40  042D  ALDP 085                             Z=G0<K20             MASK 2ND CHAR FOR 2X IN PTT CODE
1A42  E0CA  ALDP 086                  090  HINOT2 BR IF HZNZ           BR IF NOT 2X
1A44  3715  ALDP 087                             D1=D1$K10            SET CONTROL WORD FOR PROG. STORE
1A46  047B  ALDP 088                             Z=G0<K07             MASK FOR 'P' - 27
1A48  9A5E  ALDP 089                  115  BRBAD  BR                   GO TO CHECK ON '7'
1A4A  043F  ALDP 090    HINOT2                   Z=G0<K33             MASK 2ND CH. FOR C(33) OR A(31)
1A4C  E09A  ALDP 091                  044  BADCHR BR IF HZNZ           BR IF NOT 3X - INVALID 2ND CHAR
1A4E  F0D4  ALDP 095                  100  POSBUF BR IF LZNZ           GO CHECK IF DISP BUFFER OR AUX
1A50  E66C  ALDP 097                  126  ALLSET BR IF D0 BIT2=0      2ND CHAR IS 'C', BR IF DSPY CTRL
1A52  C9ED  ALDP 098                  126  ALLSET BR IF TD BIT4=1      ALTER CTRL, BR IF IN CE MODE
1A54  042B  ALDP 100    POSBUF                   Z=G0<K02             MASK FOR DISP PRT BUFF
1A56  F0DC  ALDP 101                  105  POSAUX BR IF LZNZ           BR IF 2'ND CHARACTER NOT B
1A58  E61B  ALDP 102                  044  BADCHR BR IF D02=1          BR IF 1ST CHAR A--INVAL AB
1A5A  AF78  ALDP 103                DPTC 059  START  BR                  2'ND CHAR B,DISP PRT BUFFER
1A5C  041B  ALDP 105    POSAUX                   Z=G0<K01             MASK FOR A(31) - AUX STORAGE
1A5E  F0E4  ALDP 115    BRBAD        121  POSDUM BR IF LZNZ           BR IF 2ND CH. NOT A OR P
1A60  3615  ALDP 117                            D0=D0$K10            SET CTRL WORD FOR AUX OR PROG
1A62  9A6C  ALDP 120                  126  ALLSET BR
1A64  044B  ALDP 121    POSDUM                   Z=G0<K04             TEST FOR 2ND CH. D, DUMP
1A66  F09A  ALDP 122                  044  BADCHR BR IF LZNZ           BR IF 2ND CH. NOT D
1A68  2715  ALDP 123                            D1=0$K10             SET UP DUMP FLAGS
1A6A  3625  ALDP 124                            D0=D0$K20
1A6C  6662  ALDP 126    ALLSET                   STH D DC,9C      ** STORE THE CTRL WD IN 'DOIT'(039C)
1A6E  88CC  ALDP 127                  251  SPACE  BAL                  FORCE A SPACE
1A70  88C6  ALDP 128                  247  READCH BAL                  READ THE 1ST ADDR HEX DIGIT
1A72  5403  ALDP 129                            U0=G0XH              PUT IT IN U0 HIGH
1A74  88C6  ALDP 130                  247  READCH BAL                  READ THE 2ND ADDR HEX DIGIT
1A76  440D  ALDP 131                            U0=G0L+U0H           PUT IT IN U0 LOW
1A78  88C6  ALDP 132                  247  READCH BAL                  READ THE 3RD ADDR HEX DIGIT
1A7A  5413  ALDP 133                            U1=G0XH              PUT IT IN U1 HIGH
1A7C  88C6  ALDP 134                  247  READCH BAL                  READ THE 4TH ADDR HEX DIGIT
1A7E  441D  ALDP 135                            U1=G0L+U1H           PUT IT IN U1 LOW
1A80  88BC  ALDP 136                  241  LINEFD BAL                  FORCE A LINE FEED
1A82  C310  ALDP 139                  150  ONEMOR BR IF D1 BIT4=0      BR IF CORE DUMP
1A84  F00C  ALDP 141                  145  HWDCNT BR IF G0 BIT7=0      BR IF ADDR IS EVEN
1A86  88CC  ALDP 142                  251  SPACE  BAL                  ADDR IS ODD, OFFSET BY
1A88  88CC  ALDP 143                  251  SPACE  BAL                   ONE BYTE
1A8A  17AD  ALDP 144                            D1=D1<KA0            FLIP LOOP CTRL & ODD ADDR FLAGS
1A8C  2243  ALDP 145    HWDCNT                   V0=0$K04             SET COUNTER TO 4 HW'S
1A8E  2515  ALDP 146    COLCNT                   G1=0$K10             SET COLUMN COUNT TO 16
1A90  F73D  ALDP 150    ONEMOR       202  PROGRM BR IF D1 BIT3=1      BR IF PROG STORAGE
1A92  F61F  ALDP 154                  167  EXPEOB BR IF D0 BIT3=1      48K, BR IF AUX, ALL MODULE VALID
1A94  00C9  ALDP 158                            Z=U0+KC0             MASK FOR ADDR HIGHER THAN 3FFF
1A96  F49E  ALDP 160    ADDRCK       167  EXPEOB BR IF AC=0           BR IF ADDR VALID
1A98  3713  ALDP 161    ADRCHK                   D1=D1$K01            TO TYPE 'INVALID ADDR'
1A9A  9A1C  ALDP 162                  045  BADADR BR                   ADDR OUT OF BOUNDARY
1A9C  B3B2  ALDP 165    GODUMP  DUMP 002  START  BR
1A9E  3685  ALDP 167    EXPEOB                   D0=D0$K80            SET 'EXPECTING EOB' FLAG
1AA0  E640  ALDP 168                  206  DISPLY BR IF D0 BIT2=0      BR IF DISPLAY
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
1AA2  C31C  ALDP 171                  165     GODUMP     BR IF D1 BIT4=0    BR IF CORE DUMP
1AA4  88C6  ALDP 173                  247     READCH     BAL                ALTER, READ 1ST HEX
1AA6  54F3  ALDP 174                                     H1=G0XH            PUT IT IN H1 HIGH
1AA8  88C6  ALDP 175                  247     READCH     BAL                READ 2ND HEX
1AAA  44FD  ALDP 176                                     H1=GCL+H1H         PUT IT IN H1 LOW
1AAC  172D  ALDP 177                                     D1=D1<K20          INVERT LOOPING CONTROL FLAG
1AAE  839C  ALDP 178                BCHK 330    DOIT     BAL                STORE THE BYTE
1AB0  E711  ALDP 179                  150     ONEMOR     BR IF D1 BIT2=1    BR IF NO SPACING REQUIRED
1AB2  88CC  ALDP 180  GOSPCE          251     SPACE      BAL                FORCE A SPACE
1AB4  25FF  ALDP 181  DECONT                             G1=G1+KFF          DECREMENT COLUMN COUNT BY ONE
1AB6  C490  ALDP 182                  150     ONEMOR     BR IF ZNZ          BR IF NOT 16 COLUMNS YET
1AB8  88BC  ALDP 183                  241     LINEFD     BAL                END OF ONE LINE, FORCE LINE FEED
1ABA  9A8E  ALDP 184                  146     COLCNT     BR                 GO TO RESTORE COL. COUNTER
1ABC  0049  ALDP 202  PROGRM                             Z=U0+K40      **48K, MASK FOR PROG ADDR HIGHER THAN B
1ABE  9A96  ALDP 204                  160     ADDRCK     BR
1AC0  839C  ALDP 206  DISPLY        BCHK 330    DOIT     BAL                READ HALFWORD
1AC2  1645  ALDP 207                                     D0=D0*-K40         ALLOW ANY CHAR FOR 'SPACE'
1AC4  C74A  ALDP 208                  211     ENTRY      BR IF D1 BIT0=0    BR IF ADDR IS EVEN
1AC6  1785  ALDP 209                                     D1=D1*-K80         ODD ADDR, RST ODD ADDR FLAG
1AC8  5FE9  ALDP 210  SCBYTE                             H0=H1              MOVE IN THE OTHER (ODD) BYTE
1ACA  5E35  ALDP 211  ENTRY                              V1=H0XL            MOVE HIGH HEX DIGIT INTO V1
1ACC  23FD  ALDP 212  SECHEX                             V1=V1+KF0          UNPACK HEX TO EBCDIC, ADD F0
1ACE  0361  ALDP 213                                     Z=V1+K06           TEST FOR DIGIT HIGHER THAN 9
1AD0  F4D6  ALDP 214                  217     DONUPK     BR IF AC=0         BR IF DIGIT IS 0 THRU 9
1AD2  237B  ALDP 215                                     V1=V1+K07          DIGIT IS A THRU F, CHANGE
1AD4  23CD  ALDP 216                                     V1=V1+KC0           IT TO C1 THRU C6
1AD6  88CE  ALDP 217  DONUPK          252     TYPCHR     BAL                TYPE IT OUT
1AD8  D761  ALDP 218                  222     DONE1      BR IF D1 BIT1=1    BR IF DONE ONE BYTE,LOOPING CTRL
1ADA  5E3D  ALDP 219                                     V1=H0L             MOVE IN LOW(2ND OR 4TH)HEX DIGIT
1ADC  176D  ALDP 220                                     D1=D1<K60          INVERT LOOPING CONTROL BITS
1ADE  9ACC  ALDP 221                  212     SECHEX     BR                 GO TYPE THE 2ND OR 4TH HEX DIGIT
1AE0  174D  ALDP 222  DONE1                              D1=D1<K40          INVERT LOOPING CONTROL 1
1AE2  E749  ALDP 223                  210     SCBYTE     BR IF D1 BIT2=1    BR IF JUST FINISHED 1ST BYTE
1AE4  22FF  ALDP 224                                     V0=V0+KFF          DECREMENT THE 4 HW COUNT BY 1
1AE6  C4B2  ALDP 225                  180     GOSPCE     BR IF ZNZ          BR IF COUNTER NOT ZERO YET
1AE8  3743  ALDP 226                                     D1=D1$K04          SET 16 TH CHAR FLAG
1AEA  88CE  ALDP 232                  252     TYPCHR     BAL                GO TO TYPE OUT THE LAST CHAR
1AEC  1743  ALDP 233                                     D1=D1*-K04         RST THE 16TH CHAR FLAG
1AEE  C375  ALDP 234                  237     NOBSTP     BR IF D1 BIT4=1    BR IF NOT BSTP NOR LOGOUT
1AF0  88BC  ALDP 235                  241     LINEFD     BAL                FORCE A LINE FEED
1AF2  88D2  ALDP 236                  254     NOGO       BR                 DONE INST. STEP OR LOG OUT
1AF4  2243  ALDP 237  NOBSTP                             V0=0$K04           RESTORE HW COUNTER TO 4
1AF6  88C4  ALDP 238                  246     REDCH1     BAL          **DONE 8 BYTES, GO GET EOB OR ANY CHAR
1AF8  9AB4  ALDP 239                  181     DECONT     BR                 NOT EOB, DISPLAY 8 MORE BYTES
            ALDP 240  AEND
08BC  2315  ALDP 241   LINEFD                            V1=0$K10           LINE FEED SUBROUTINE
08BE  3353  ALDP 242                                     V1=V1$K05          SET V1 TO LINE FEED CHARACTER
08C0  3F00  ALDP 243                                     SET TA K=80        ALLOW KEYBOARD RESTORE
08C2  C4CE  ALDP 244                  252     TYPCHR     BR IF ZNZ          UNCONDITIONAL BR
08C4  2F20  ALDP 246   REDCH1                            SET TA K=02        INITIALIZE PRINTER
08C6  3F02  ALDP 247   READCH                            SET TA K=90        SET READ LATCH, & ALDP ACTIVE
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS
08C8  1683  ALDP 248                                       D0=D0*-K08         SET READ CHAR SUBROUTINE FLAG
08CA  C4D0  ALDP 249                  253        INTVCK     BR IF ZNZ          UNCONDITIONAL BR
08CC  2345  ALDP 251  SPACE                                 V1=0$K40           SPACE SUBROUTINE, V1=BLANK
08CE  2F0A  ALDP 252  TYPCHR                                SET TA K=50        TYPE CHAR ROUTINE, SET WRITE LTH
08D0  EEA2  ALDP 253  INTVCK          262        STRADR     BR IF TT BIT2=0    BR IF NO INTERVENTION REQUIRED
08D2  1F2A  ALDP 254  NOGO                                  RST TA K=D2        RST RD, WT, ACTIVE & SHARE
08D4  E31A  ALDP 255                  258        ADDED      BR IF D1 BIT6=0    BR IF NORMAL ALDP
08D6  FA9B  ALDP 256                  258        ADDED      BR IF TT BIT7=1    BR IF LOGOUT LATCH ON
08D8  9302  ALDP 257                BCHK 069     ENTRY      BR                 CK SUM LOGOUT DONE
089A  2400  ALDP 258  ADDED                                  SET MODE K=00      SET CPU MODE TO BR ON MC REG
089C  AAB1  ALDP 259                  261        NOMCK   N  BR IF MC=NZ        BR IF LOGOUT TO BE DONE YET
089E  0610  ALDP 260                                        RST BC K=01        RST LOGOUT LATCH
08A0  AB20  ALDP 261  NOMCK   0     BSWI 107     RESTRH     BR                 GO TO SOFT STOP
08A2  3BE9  ALDP 262  STRADR                                T1=0-KE0           T=001F
08A4  76AA  ALDP 263                                        STH D AS,T-2       STORE D(CTRL WORD OR FLAGS)1E-1F
08A6  72AA  ALDP 264                                        STH V AS,T-2       V(HW COUNT & WRITE DATA)IN 1C-1D
08A8  74AA  ALDP 265                                        STH G AS,T-2       STORE G (COLLUMN COUNT) IN 1A-1B
08AA  70A0  ALDP 266                                        STH U AS,T+0       STORE U(ADDR REG'S) IN 0018-0019
08AC  2B2D  ALDP 267                                        T1=T1+K20          T=0039
08AE  7EA0  ALDP 268                                        STH H AS,T+0       STORE H(DATA REG'S) IN 0038-0039
08B0  2440  ALDP 269                                        SET MODE K=04      SET TO BACK UP ZONE
08B2  7872  ALDP 270                                        STH I DA,9E        STORE THE BAL RETURN ADDR IN K-7
08B4  2400  ALDP 271  RESET                                 SET MODE K=00      SET TO CPU MODE & ZONE
08B6  E33B  ALDP 272                  274        GOBSWI     BR IF D1 BIT6=1    BR IF FROM 'BMCK' ROUTINE
08B8  1600  ALDP 273                                        RST BC K=80        SET SOFT STOP LATCH
08BA  AB20  ALDP 274  GOBSWI    BSWI 107     RESTRH     BR                 GO TO SOFT STOP
26B8  3BE9  ALDP 275  RTTYP                                 T1=0-KE0           T=001F
26BA  56AA  ALDP 276                                        RDH D AS,T-2       RESTORE D(CTRL WORD OR FLAGS)
26BC  52AA  ALDP 277                                        RDH V AS,T-2     **RESTORE V(HW COUNTER & WRITE DATA)
26BE  54AA  ALDP 278                                        RDH G AS,T-2       RESTORE G(COL. COUNT)
26C0  42AF  ALDP 279                                        STPO=TO            DISABLE STORAGE PROTECT
26C2  C218  ALDP 280                  291        RDCHRT     BR IF D0 BIT4=0    BR IF IT WAS FROM 'READCH'
26C4  D349  ALDP 281                  283        BAKMOD     BR IF D1 BIT5=1    BR IF IT WAS THE 16TH CHAR
26C6  4F3F  ALDP 282                                        TE=V1              SEND CHAR TO BUS OUT
26C8  2440  ALDP 283  BAKMOD                                SET MODE K=04      SET TO BACK UP MODE
26CA  5872  ALDP 284                                        RDH I DA,9E        RESTORE BAL RETURN ADDR
26CC  2404  ALDP 285                                        SET MODE K=20      SET BACK TO 1052 MODE
26CE  50A0  ALDP 286                                        RDH U AS,T+0       RESTORE U(ADDR REG'S)
26D0  2B2D  ALDP 287                                        T1=T1+K20          T=0039
26D2  5EA0  ALDP 288                                        RDH H AS,T+0       RESTORE H(DATA REG'S)
26D4  1F28  ALDP 289                                        RST TA K=C2        RST READ, WRITE LATCH, SHARE REQ
26D6  128E  ALDP 290                                        RTN
2698  5A4F  ALDP 291  RDCHRT                                GO=TI              GET CHAR. FROM BUS IN
269A  3683  ALDP 292                                        D0=D0$K08          RST 'READCH' BAL ROUTINE FLAG
269C  CAAF  ALDP 293                  302        RSTLCH     BR IF TT BIT4=1    BR IF KEY BOARD CK
269E  14C5  ALDP 294                                        GO=GO*-KC0         ALLOW BOTH UPPER & LOWER CASES
26A0  DAB2  ALDP 295                  304        NOEOBC     BR IF TT BIT5=0    BR IF NOT ALTERNATE CODE
26A2  04AB  ALDP 296                                        Z=GO<K0A           MASK FOR 'CANCEL'
26A4  C4AF  ALDP 297                  302        RSTLCH     BR IF Z=0          BR IF 'CANCEL'
26A6  1623  ALDP 298                                        D0=D0*-K02         SET INVALID CHAR. FLAG
26A8  045B  ALDP 299                                        Z=GO<K05           MASK FOR EOB
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

26AA    C4B2    ALDP 300                  304     NOEOBC     BR IF ZNZ          BR IF NOT EOB
26AC    C632    ALDP 301                  304     NOEOBC     BR IF DO BIT0=0    BR IF NOT EXPECTING EOB
26AE    1F28    ALDP 302    RSTLCH                           RST TA K=C2        RST READ, WRITE LATCH, SHARE REQ
26B0    9A0C    ALDP 303                  037     START      BR                 RESTART ANOTHER PASS
26B2    957A    ALDP 304    NOEOBC    DYPE 242     STTRSL     BAL                GO TO X'LATE CHAR IN H1
26B4    E201    ALDP 305                  307     NOIVLD     BR IF DO BIT6=1    BR IF INVALID CH. FLAG IS NOT ON
26B6    9A1A    ALDP 306    GOBAD         044     BADCHR     BR                 CHAR. TYPED IS INVALID
2680    D648    ALDP 307    NOIVLD        283     BAKMOD     BR IF DO BIT1=0    BR IF PACKING IS NOT REQUIRED
2682    CFFD    ALDP 308                                     Z=H1<KF0           MASK FOR NUMERIC
2684    E095    ALDP 309                  317     DONPCK     BR IF HZ=0         BR IF CH. IS NUMERIC
2686    1FCD    ALDP 310                                     H1=H1<KC0          INVERT HIGH TWO BITS
2688    E08F    ALDP 311                  314     ALPHA      BR IF HZ=0         BR IF CX
268A    1F4D    ALDP 312                                     H1=H1<K40          MASK FOR 8X
268C    E0B6    ALDP 313                  306     GOBAD      BR IF HZNZ         BR IF CHAR IS NOT CX NOR 8X
268E    F0B7    ALDP 314    ALPHA         306     GOBAD      BR IF LZ=0         BR IF CHAR IS C0 OR 80 - INVALID
2690    2F9B    ALDP 315                                     H1=H1+K09          ADJUST
2692    E0B6    ALDP 316                  306     GOBAD      BR IF HZNZ         BR IF CH. IS NOT A THRU F
2694    5F4D    ALDP 317    DONPCK                           GO=H1L             DONE PACKING
2696    A6C8    ALDP 318                  283     BAKMOD     BR
                ALDP 319    ATABLE  ADDR=0044
0044    D9C4    ALDP 320    C                                CAUX'RDDA'
                ALDP 321    AEND
                ALDP 322    ATABLE  ADDR=0054
0054    D9C1    ALDP 323    C                                CAUX'RAHC'
                ALDP 324    AEND
                ALDP 325    ATABLE  ADDR=0064
0064    40C4    ALDP 326    C                                CAUX' DIL'
                ALDP 327    AEND
                ALDP 328    ATABLE  ADDR=0074
0074    C1E5    ALDP 329    C                                CAUX'AVNI'
                ALDP 330    AEND
```

```
                                    **********************************
                                    * CROSS REFERENCE FOR CSECT ALDP *
                                    **********************************

ALDP 037    ALDP 050   ALDP 303   DYPE 019
ALDP 044    ALDP 059   ALDP 091   ALDP 102   ALDP 122   ALDP 306
ALDP 045    ALDP 162
ALDP 047    ALDP 049   ALDP 052   ALDP 054
ALDP 051    ALDP 055
ALDP 056    ALDP 043
ALDP 061    ALDP 057
ALDP 090    ALDP 086
ALDP 100    ALDP 095
ALDP 105    ALDP 101
ALDP 115    ALDP 089
ALDP 121    ALDP 115
ALDP 126    ALDP 097   ALDP 098   ALDP 120
ALDP 145    ALDP 141   BCHK 068
ALDP 146    ALDP 184   BMCK 133
ALDP 150    ALDP 139   ALDP 179   ALDP 182
ALDP 160    ALDP 204
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT ALDP *
                                  **********************************
ALDP 165    ALDP 171
ALDP 167    ALDP 154  ALDP 160
ALDP 180    ALDP 225
ALDP 181    ALDP 239
ALDP 202    ALDP 150
ALDP 206    ALDP 168
ALDP 210    ALDP 223
ALDP 211    ALDP 208  BSTP 025
ALDP 212    ALDP 221
ALDP 217    ALDP 214
ALDP 222    ALDP 218
ALDP 237    ALDP 234
ALDP 241    ALDP 039  ALDP 044  ALDP 136  ALDP 183  ALDP 235  BMCK 132  DPTC 063
ALDP 246    ALDP 040  ALDP 238
ALDP 247    ALDP 061  ALDP 128  ALDP 130  ALDP 132  ALDP 134  ALDP 173  ALDP 175
ALDP 251    ALDP 127  ALDP 142  ALDP 143  ALDP 180  DPTC 087
ALDP 252    ALDP 048  ALDP 217  ALDP 232  ALDP 244  DPTC 081
ALDP 253    ALDP 249
ALDP 254    ALDP 236
ALDP 258    ALDP 255  ALDP 256
ALDP 261    ALDP 259
ALDP 262    ALDP 253
ALDP 271    BSWI 175
ALDP 274    ALDP 272
ALDP 275    DYPE 020
ALDP 283    ALDP 281  ALDP 307  ALDP 318
ALDP 291    ALDP 280
ALDP 302    ALDP 293  ALDP 297  DPTC 095
ALDP 304    ALDP 295  ALDP 300  ALDP 301
ALDP 306    ALDP 313  ALDP 314  ALDP 316
ALDP 307    ALDP 305
ALDP 314    ALDP 311
ALDP 317    ALDP 309
```

BCHK DESCRIPTIVE TEXT

THE CHECK SUM ROUTINE -BCHK-, IS ENTERED UPON COMPLETION OF THE RESIDENT CPU MICRODIAGNOSTIC -BDIA-. THESE ROUTINES ARE EXECUTED WHENEVER THE CSL, SYSTEM RESET, OR LOAD KEYS ARE PRESSED.

THE CHECK SUM ROUTINE PERFORMS AN EXCLUSIVE OR ON THE CONTENTS OF CONTROL STORAGE WITH THE EXCEPTION OF LOCATIONS 0002-000D AND 0280-02BF.

LOCATIONS 0002-000D CONTAIN THE HANDLOADED CHECK SUM VALUES OF THE INDIVIDUAL CORE LOADS. WHEN A CSL OPERATION IS PERFORMED, THE CHECK SUM VALUE IS SELECTED FROM THIS AREA AND PLACED IN LOCATION 0EC4, WHERE IT BECOMES PART OF THE AREA THAT IS SUBJECTED TO THE EXCLUSIVE OR. THIS CHECK SUM VALUE SHOULD CAUSE THE EXCLUSIVE OR RESULT TO BE ZERO.

THE HANDLOADED AREAS ARE -

ADDRESS	CHECK SUM VALUE FOR
0002	*E60/*E61 (MODEL 25 MODE)
0004	*E62
0006	*E63
0008	*E40 (1401/1460 MODE)
000A	*E50 (1440 MODE)
000C	*E20 (MODEL 20 MODE)

THE CHECK SUM ROUTINE IS EXECUTED IN THE FOLLOWING SEQUENCE-

* 1 * ENTRY IS MADE FROM THE BDIA ROUTINE TO LABEL -START-
* 2 * SET 1052 MODE AND INITIALIZE CERTAIN DIRECT ADDRESSABLE CONTROL STORAGE LOCATIONS.
* 3 * TEST LOGOUT- *ON * A CSL OPERATION HAD BEEN PERFORMED. SET UP THE LOGOUT AREA WITH CHECK SUM INFORMATION.
  0EC0 -- EC LEVEL
  0EC2 -- LABEL OF CORE LOAD BEING USED.
  0EC4 -- VALUE FROM HANDLOAD AREA.
  0EC6 -- CORRECTION FACTOR, ALWAYS ZERO ON INITIAL LOGOUT.
  GO TO -ALDP- ROUTINE TO LOGOUT THIS INFORMATION. RETURN IS MADE TO -BCHK- AT LABEL * ENTRY *.
  LOGOUT *OFF* ENTRY WAS MADE AFTER A SYSTEM RESET OR LOAD OPERATION HAD BEEN PERFORMED. GO TO -BCHK- LABEL * ENTRY *.
* 4 * PERFORM EXCLUSIVE OR OF CONTROL STORAGE. SKIP LOCATIONS 0002-000D AND 0280-02BF.
* 5 * CHECK SUM OK - BRANCH TO -BSYS- LABEL * SYSRST *
  CHECK SUM BAD -SET UP ADDRESS OF CORRECTION FACTOR PRIOR TO CE KEY TEST. GO TO STEP 6.
* 6 * CE KEY ON - STORE EXCLUSIVE OR RESULT AS CORRECTION FACTOR INTO ADDRESS 0EC6. BRANCH TO LOGOUT CHECK SUM AREA AND PERFORM AN EXCLUSIVE OR AGAIN. THIS SHOULD RESULT IN A GOOD CHECK SUM.
  CE KEY OFF- TURN ON CSL CHECK LIGHT AND ** STOP **. (HARD STOP LATCH ON)

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BCHK 001	T		CHECK SUM		
		BCHK 002	*				
		BCHK 003	*				
		BCHK 004	*				
1324	2404	BCHK 011	START			SET MODE K=20	SET 1052 MODE
1326	4632	BCHK 051				RDH D 0C,8E	SET T TO 4E08
1328	6662	BCHK 052				STH D 0C,9C	STORE 4E08 INTO 'DOIT' WORD
132A	FA82	BCHK 053		069	ENTRY	BR IF TT7=0	FALL THRU IF LOGOUT ON(CSL ONLY)
132C	21C5	BCHK 054	AGAIN			U1=0$KC0	BUILD ADDRESS OF CORE
132E	20E3	BCHK 055				U0=0$K0E	LOAD NAME JUST LOADED
1330	6004	BCHK 056				U=U+2	U-REGISTER = 0EC2
1332	4208	BCHK 057				RDH V CS,U+2	READ OUT THE NAME
1334	2A07	BCHK 058				T0=0	
1336	53BD	BCHK 059				T1=V1L	FIND OUT WHAT
1338	C4BC	BCHK 060		062	CALARE	BR IF ZNZ	THE CSL NAME IS
133A	2B1B	BCHK 061				T1=T1+K01	ADD 1 TO LOW DIGIT OF NAME
133C	6BB3	BCHK 062	CALARE			T1=T1+T1	DOUBLE THE LOW DIGIT OF NAME

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
133E  42A0  BCHK 063                                    RDH V CS,T     ** USE UPDATED NAME TO READ HANDLOAD
1340  6208  BCHK 064                                    STH V CS,U+2       STORE HANDLOAD CHECK SUM IN 0EC4
1342  0610  BCHK 065                                    RST BC K=01        RESET LOGOUT LATCH
1344  511B  BCHK 066                                    U1=U1H             U-REG=0EC0
1346  2723  BCHK 067                                    D1=0$K02           SET UP FLAGS
1348  9A8C  BCHK 068          ALDP 145  HWDCNT           BR                 BRANCH TO LOGOUT 0EC0--0EC6
1302  4252  BCHK 069  ENTRY                              RDH V DC,9A        ZERO V-REGISTER
1304  4C26  BCHK 070                                    P=V                ZERO P-REGISTER
1306  4AC0  BCHK 071                                    RDH T CS,P         READ OUT CS LOCATION 0000
1308  2DEB  BCHK 072                                    P1=P1+K0E          SKIP HANDLOAD INFORMATION
130A  8258  BCHK 073  MORE          257  DOFUNC        BAL                BR TO START EXCLUSIVE OR
130C  6CC4  BCHK 074                                    P=P+2              INCREMENT CS ADDRESS
130E  DC0A  BCHK 083               073  MORE          BR IF P01=0
1310  42A6  BCHK 092                                    V=T                MOVE LAST EX OR RESULT TO  V
1312  C481  BCHK 093               105  BREAK         BR IF Z=0          BR IF EX OR RESULT ZERO
1314  2CE3  BCHK 094                                    P0=0$K0E           SET UP
1316  2DC5  BCHK 095                                    P1=0$KC0              CORRECTION
1318  3D63  BCHK 096                                    P1=P1$K06                FACTOR ADDRESS
131A  C99F  BCHK 097               100  OBOY          BR IF TD4=1        BRANCH IF CE KEY ON
131C  2820  BCHK 098                                    SET DR K=02        TURN ON CHECK SUM LIGHT AND THE
            BCHK 099  *                                                    HARD STOP LATCH.
131E  8258  BCHK 100   OBOY        257  DOFUNC        BAL                GO GENERATE CORRECTION FACTOR
1320  6AC0  BCHK 101                                    STH T CS,P         STORE CORRECTION FACTOR IN 0EC6
1322  932C  BCHK 102               054  AGAIN         BR                 GO SETUP FOR CORRECTION LOGOUT
24F8  4A52  BCHK 103   OKDONE                          RDH T DC,9A        ZERO OUT T REG
24FA  A28C  BCHK 104          BSYS 002  SYSRST         BR                 BR TO SYSTEM RESET
1300  A4E6  BCHK 105   BREAK         107  OKSUM         BR                 STRING BREAK BR
24E6  26C5  BCHK 107   OKSUM                           D0=0$KC0           SET UP AUX MOD C
24E8  2705  BCHK 127                                    D1=0$K00
24EA  4452  BCHK 128   SETUP                           RDH G DC,9A        SET G TO 00
24EC  4248  BCHK 129   RD                              RDH V CS,G+2       READ CTRL AND
24EE  7268  BCHK 130                                    STH V AS,D+2       STORE INTO AUX
24F0  F06C  BCHK 131                    129  RD        BR IF G07=0        BR IF NOT DONE
24F2  F679  BCHK 175                    103  OKDONE    BR IF D03=1        BR IF DONE
24F4  26D5  BCHK 192                                    D0=0$KD0           SET UP AUX MOD D
24F6  A4EA  BCHK 210                    128  SETUP     BR                 BR TO STORE OTHER MOD C
0258  0C2B  BCHK 257   DOFUNC                          Z=P0<K02           BYPASS
025A  C4E4  BCHK 258                    263  NOTCE     BR IF ZNZ             CE
025C  0D8D  BCHK 259                                    Z=P1<K80                 TRAP
025E  C4E4  BCHK 260                    263  NOTCE     BR IF ZNZ                   AREA
0260  2DB5  BCHK 261                                    P1=0$KB0                      0280-02BE
0262  2DEB  BCHK 262                                    P1=P1+K0E
0264  42C0  BCHK 263   NOTCE                           RDH V CS,P         READ FROM CONTROL STORAGE
0266  6A21  BCHK 264                                    T0=T0<V0           PERFORM THE
0268  6B31  BCHK 265                                    T1=T1<V1              EXCLUSIVE OR
026A  128E  BCHK 266                                    RTN
            BCHK 267   RESERVE 0280 THRU 02BC
            BCHK 269   ATABLE  ADDR=02BE
02BE  B0EC  BCHK 270                OVLY 002  IXSTRT     BR
            BCHK 271   AEND
            BCHK 315   RESERVE 0EC0 THRU 0EC2
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT              COMMENTS

                 BCHK 320    ATABLE    ADDR=0EC4
0EC4     0000    BCHK 321    C                                 XCTL'00000000'
                 BCHK 322    AEND
                 BCHK 323    ATABLE    ADDR=0388
0388     07FF    BCHK 324    C                                 XCTL'07FFC48A88004E08'
                 BCHK 325    AEND
                 BCHK 326    ATABLE    ADDR=0398
0398     00C8    BCHK 327    C                                 XCTL'00C80000'
                 BCHK 328    AEND
                 BCHK 329    ATABLE    ADDR=039C
039C     4E08    BCHK 330    CDOIT                             XCTL'4E08128E'
                 BCHK 331    AEND
                 BCHK 332    ATABLE    ADDR=03A8
03A8     0000    BCHK 334    C                                 XCTL'0000000000000000'
                 BCHK 347    AEND
                 BCHK 348    ATABLE    ADDR=03B8
03B8     2FF7    BCHK 349    C                                 XCTL'2FF72F1BC4BC0F01'
                 BCHK 350    AEND
                                       ************************************
                                       * CROSS REFERENCE FOR CSECT BCHK *
                                       ************************************

BCHK 011     BDIA 432
BCHK 054     BCHK 102
BCHK 062     BCHK 060
BCHK 069     ALDP 257   BCHK 053
BCHK 073     BCHK 083
BCHK 100     BCHK 097
BCHK 103     BCHK 175
BCHK 105     BCHK 093
BCHK 107     BCHK 105
BCHK 128     BCHK 210
BCHK 129     BCHK 131
BCHK 257     BCHK 073   BCHK 100
BCHK 263     BCHK 258   BCHK 260
BCHK 330     ALDP 178   ALDP 206
```

BCPL DESCRIPTIVE TEXT

THE BCPL ROUTINE IS NORMALLY RESIDENT IN CONTROL STORAGE, AND IS USED TO LOAD THE INITIAL RECORD OF EITHER THE CHANNEL OR NATIVE BOOTSTRAP ROUTINES.

IF THE BCPL ROUTINE HAS BEEN ALTERED, THE APPROPRIATE HANDLOAD INFORMATION MUST BE ENTERED TO BE SURE OF CORRECT CSL OPERATION.

*** SWITCH SETTINGS FOR CSL ***

SWITCHS

A,B = CC	CSL FROM CHANNEL
A,B = DD	CSL FROM NATIVE 2560
A,B = EE	CSL FROM NATIVE 2540
A,B = FF	CSL FROM NATIVE 2311

SWITCHES C,D SET TO ACTUAL UNIT ADDRESS

PROCEDURE FOR PUNCHING CSL CARDS

COLUMNS	
1-2	CONTAIN STARTING ADDRESS TO BE LOADED
3	CONTAINS CODE INFORMATION HEX 80 = DATA IS FOR PROGRAM STORAGE HEX 40 = LAST CSL CARD OF DECK HEX 20 = DATA IS FOR AUXILIARY STORAGE HEX 10 = DATA IS FOR CONTROL STORAGE
4	CONTAINS THE NUMBER OF HALFWORDS TO BE LOADED —
5-68	CONTAIN THE DATA
69-72	OPTIONAL, CAN BE USED FOR ANY INFORMATION.
73-77	*NNN9 WHERE N IS THE CORE LOAD ID.
78-80	XXX 3 DIGIT NUMBER INDICATING SEQ. OF PATCHES.

*** WARNING- DO NOT PUNCH CARDS TO LOAD INTO AUX MODULE 1,3,4

RESTRICTIONS WHEN PUNCHING CSL CARDS

COLS 1-2 ALL ADDRESSES SHOULD BE EVEN.
CONTROL STORAGE - THESE ADDRESSES SHOULD BE IN THE RANGE OF 0000-3FFE ONLY.
AUXILIARY STORAGE- THE 2ND HEX CHARACTER IN COLUMN 1 SHOULD BE A ZERO.
PROGRAM STORAGE - THE ADDRESS MUST BE IN THE RANGE OF THE SYSTEM.

COL 3 CODED INFORMATION

COL 4 COUNT FIELD - IF A COUNT OF 0 IS INDICATED, 257 HALFWORDS WILL BE LOADED.

COLS 5-68 DATA FIELD - CAN BE ANYTHING

COLS 69-80 SEE ABOVE.
NOT USED BY THE CSL.

*** WHEN LOADING A FULL DECK WITH REPLACE CARDS, THE REPLACE CARDS SHOULD GO JUST BEFORE THE END CARD.

*** WHEN LOADING ONLY REPLACE CARDS, IT IS NECESSARY TO PUT THE CONTROL STORAGE CARDS (NORMALLY 4) FOR MODULE 01XX IN THE DECK.
ORDER OF THE CARDS IS - BOOTSTRAP CARDS, CONTROL STORAGE CARDS FOR MODULE 01XX, REPLACE CARDS, END CARD.

** EXAMPLE OF PUNCHING AND LOADING A REPLACE CARD **

ASSUME THAT THE DEVICE ADDRESS FOR THE 1403 ON THE BURST CHANNEL NEEDED TO BE CHANGED TO -0A-.

THE STANDARD ADDRESS FOR THE 1403 ON THE BURST CHANNEL IS -0E- AND IS LOCATED IN AUXILIARY STORAGE MODULE 0, AT ADDRESS -87-.

THE CARD TO CHANGE THIS WOULD BE PUNCHED IN THE FOLLOWING MANNER--

COL 1	(HEX 00)	THIS ADDRESSES MODULE 0.
COL 2	(HEX 86)	ALTHOUGH THE CHANGE IS FOR ADDRESS 87, THE HEX ADDRESS REPRESENTED BY COLUMN 2 MUST BE EVEN.
COL 3	(HEX 20)	THIS INDICATES INFORMATION IS FOR AUXILIARY STORAGE.
COL 4	(HEX 01)	THIS INDICATES ONE HALFWORD TO BE LOADED.
COL 5	(HEX 0D)	THIS REPRESENTS THE DEVICE ADDRESS OF THE 2540 PUNCH THAT MUST BE RELOADED BECAUSE OF THE ADDRESSING RESTRICTION IMPOSED BY COLUMN 2.
COL 6	(HEX 0A)	THIS REPRESENTS THE DEVICE ADDRESS
COL 69-72	(0086)	TO INDICATE START ADDR OF PATCH.
COL 73-77	(*E609)	INDICATES PATCH IS FOR *E60 CORE LD.
COL 78-80	(001)	INDICATES 1ST PATCH TO DECK REPLACING THE STANDARD ADDRESS FOR THE 1403 ON THE BURST CHANNEL.

A CSL MUST BE PERFORMED TO LOAD THIS RECONFIGURATION CARD INTO THE SYSTEM. THE RECONFIGURATION CARD MUST PRECEED THE END CARD, AND MUST BE PRESENT IN THAT POSITION FOR ALL CSLS USING THAT PARTICULAR CORE LOAD DECK.

*** EACH TIME A NEW EC LEVEL DECK IS RECIEVED, ALL RECONFIGURATION CARDS IN THE DECK BEING REPLACED MUST BE REMOVED AND INSERTED IN THE NEW DECK IN FRONT OF THE END CARD.

PUNCHED CARD CODES

HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES	HEX	PUNCHES
00	T-0-9-8-1	25	0-9-5	4A	T-8-2	6F	0-8-7	94	T-E-4	B9	T-E-0-9	DE	T-E-9-8-6
01	T-9-1	26	0-9-6	4B	T-8-3	70	T-E-0	95	T-E-5	BA	T-E-0-8-2	DF	T-E-9-8-7
02	T-9-2	27	0-9-7	4C	T-8-4	71	T-E-0-9-1	96	T-E-6	BB	T-E-0-8-3	E0	0-8-2
03	T-9-3	28	0-9-8	4D	T-8-5	72	T-E-0-9-2	97	T-E-7	BC	T-E-0-8-4	E1	E-0-9-1
04	T-9-4	29	0-9-8-1	4E	T-8-6	73	T-E-0-9-3	98	T-E-8	BD	T-E-0-8-5	E2	0-2
05	T-9-5	2A	0-9-8-2	4F	T-8-7	74	T-E-0-9-4	99	T-E-9	BE	T-E-0-8-6	E3	0-3
06	T-9-6	2B	0-9-8-3	50	T	75	T-E-0-9-5	9A	T-E-8-2	BF	T-E-0-8-7	E4	0-4
07	T-9-7	2C	0-9-8-4	51	T-E-9-1	76	T-E-0-9-6	.B	T-E-8-3	C0	T-0	E5	0-5
08	T-9-8	2D	0-9-8-5	52	T-E-9-2	77	T-E-0-9-7	9C	T-E-8-4	C1	T-1	E6	0-6
09	T-9-8-1	2E	0-9-8-6	53	T-E-9-3	78	T-E-0-9-8	9D	T-E-8-5	C2	T-2	E7	0-7
0A	T-9-8-2	2F	0-9-8-7	54	T-E-9-4	79	8-1	9E	T-E-8-6	C3	T-3	E8	0-8
0B	T-9-8-3	30	T-E-0-9-8-1	55	T-E-9-5	7A	8-2	9F	T-E-8-7	C4	T-4	E9	0-9
0C	T-9-8-4	31	9-1	56	T-E-9-6	7B	8-3	A0	E-0-8-1	C5	T-5	EA	E-0-9-8-2
0D	T-9-8-5	32	9-2	57	T-E-9-7	7C	8-4	A1	E-0-1	C6	T-6	EB	E-0-9-8-3
0E	T-9-8-6	33	9-3	58	T-E-9-8	7D	8-5	A2	E-0-2	C7	T-7	EC	E-0-9-8-4
0F	T-9-8-7	34	9-4	59	E-8-1	7E	8-6	A3	E-0-3	C8	T-8	ED	E-0-9-8-5
10	T-E-9-8-1	35	9-5	5A	E-8-2	7F	8-7	A4	E-0-4	C9	T-9	EE	E-0-9-8-6
11	E-9-1	36	9-6	5B	E-8-3	80	T-0-8-1	A5	E-0-5	CA	T-0-9-8-2	EF	E-0-9-8-7
12	E-9-2	37	9-7	5C	E-8-4	81	T-0-1	A6	E-0-6	CB	T-0-9-8-3	F0	0
13	E-9-3	38	9-8	5D	E-8-5	82	T-0-2	A7	E-0-7	CC	T-0-9-8-4	F1	1
14	E-9-4	39	9-8-1	5E	E-8-6	83	T-0-3	A8	E-0-8	CD	T-0-9-8-5	F2	2
15	E-9-5	3A	9-8-2	5F	E-8-7	84	T-0-4	A9	E-0-9	CE	T-0-9-8-6	F3	3
16	E-9-6	3B	9-8-3	60	E	85	T-0-5	AA	E-0-8-2	CF	T-0-9-8-7	F4	4
17	E-9-7	3C	9-8-4	61	0-1	86	T-0-6	AB	E-0-8-3	D0	E-0	F5	5
18	E-9-8	3D	9-8-5	62	E-0-9-2	87	T-0-7	AC	E-0-8-4	D1	E-1	F6	6
19	E-9-8-1	3E	9-8-6	63	E-0-9-3	88	T-0-8	AD	E-0-8-5	D2	E-2	F7	7
1A	E-9-8-2	3F	9-8-7	64	E-0-9-4	89	T-0-9	AE	E-0-8-6	D3	E-3	F8	8
1B	E-9-8-3	40	NONE	65	E-0-9-5	8A	T-0-8-2	AF	E-0-8-7	D4	E-4	F9	9
1C	E-9-8-4	41	T-0-9-1	66	E-0-9-6	8B	T-0-8-3	B0	T-E-0-8-1	D5	E-5	FA	T-E-0-9-8-2
1D	E-9-8-5	42	T-0-9-2	67	E-0-9-7	8C	T-0-8-4	B1	T-E-0-1	D6	E-6	FB	T-E-0-9-8-3
1E	E-9-8-6	43	T-0-9-3	68	E-0-9-8	8D	T-0-8-5	B2	T-E-0-2	D7	E-7	FC	T-E-0-9-8-4
1F	E-9-8-7	44	T-0-9-4	69	0-8-1	8E	T-0-8-6	B3	T-E-0-3	D8	E-8	FD	T-E-0-9-8-5
20	E-0-9-8-1	45	T-0-9-5	6A	T-E	8F	T-0-8-7	B4	T-E-0-4	D9	E-9	FE	T-E-0-9-8-6
21	0-9-1	46	T-0-9-6	6B	0-8-3	90	T-E-8-1	B5	T-E-0-5	DA	T-E-9-8-2	FF	T-E-0-9-8-7
22	0-9-2	47	T-0-9-7	6C	0-8-4	91	T-E-1	B6	T-E-0-6	DB	T-E-9-8-3		
23	0-9-3	48	T-0-9-8	6D	0-8-5	92	T-E-2	B7	T-E-0-7	DC	T-E-9-8-4		
24	0-9-4	49	T-8-1	6E	0-8-6	93	T-E-3	B8	T-E-0-8	DD	T-E-9-8-5		

*** HANDLOAD ROUTINE FOR NATIVE 2540 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	P0=0	ZERO OUT SWITCH
0016	2413	G0=0$K01	START SETUP OF ADDR 0100
0018	8062	BR	BRANCH TO LOCATION 0062
0062	2406	SET MODE K=30	SET 2540 MODE
0064	2617	D0=0$K11	BUILD AUXILIARY
0066	2783	D1=0$K08	STORAGE ADDRESS
0068	3775	D1=D1$K70	FOR COMPARING
006A	2507	G1=0	FINISH SETUP OF ADDR 0100
006C	4066	U=D	SAVE COMPARE ADDRESS
006E	5EEF	H0=RPS	READ 2540 STATUS
0070	DE6E	BR IF H01=0	BACK ONE WORD IF NO DATA
0072	5B3F	V1=RP1	GET READ BRUSH 1 DATA
0074	6348	STB V1 CS,G+1	PUT DATA IN CONTROL STOR
0076	736A	STB V1 AS,D-1	PUT DATA IN COMP TABLE
0078	F0EE	BR IF LZNZ	GO WAIT FOR MORE DATA
007A	161B	D0=D0<K01	INVERT FIRST TIME SWITCH
007C	F0E7	BR IF LZ=0	BRANCH BACK IF FIRST TIME
007E	8100	BR	BRANCH TO BOOTSTRAP

*** HANDLOAD ROUTINE FOR CHANNEL ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	P0=0	ZERO OUT SWITCH
0016	2413	G0=0$K01	START SETUP OF ADDR 0100
0018	51AF	T0=SWCD	SWCD EQUAL DEVICE ADDR.
001A	802C	BR	BRANCH TO LOCATION 002C
002C	2486	SET MODE K=38	SET CHANNEL MODE,CPU ZONE
002E	2507	G1=0	FINISH SETUP OF ADDR 0100
0030	2B08	SET GA K=40	SET SERVICE OUT
0032	C9B3	BR IF GT4=1	BRANCH ON OP IN
0034	4FAF	GB/OUT=T0	SEND DEVICE ADDRESS OUT
0036	2B04	SET GA K=20	RAISE ADDRESS OUT
0038	2B44	SET GA K=24	AND SELECT OUT
003A	C9BA	BR IF GT4=0	WAIT FOR OP IN
003C	2B40	SET GA K=04	RESET ADDRESS OUT
003E	CDBE	BR IF GT0=0	WAIT FOR ADDRESS IN
0040	2B23	T1=0$K02	BUILD READ COMMAND
0042	4FBF	GB/OUT=T1	SEND OUT READ COMMAND
0044	2B42	SET GA K=14	RAISE COMMAND OUT
0046	FDC6	BR IF GT3=0	WAIT HERE FOR STATUS
0048	5FBF	T1=GB/IN	READ STATUS
004A	C4CA	BR IF ZNZ	LOOP HERE IF INVALID STAT
004C	2B48	SET GA K=44	SET SERVICE OUT
004E	FDCF	BR IF GT3=1	WAIT FOR
0050	EDCE	BR IF GT2=0	DATA
0052	5FFF	H1=GB/IN	GET DATA BYTE
0054	F058	BR IF G07=1	BR IF BOOTSTRAP READING
0056	7F48	STB H1 AS,G+1	NOT BOOT INFO,STOR IN AUX
0058	F05C	BR IF G07=0	UNCONDITIONAL BRANCH
005A	6F48	STB H1 CS,G+1	PUT BOOT IN CNTRL STORAGE
005C	055D	Z=G1<K50	CHECK IF ALL DATA IN,
005E	C4CC	BR IF ZNZ	IF NOT, GET MORE.
0060	8100	BR	BRANCH TO BOOTSTRAP

********THE FOLLOWING TECHNIQUE MAY BE USED TO ENTER DATA INTO SEQUENTIAL MEMORY ADDRESSES WHEN ALTER/DISPLAY IS NOT AVAILABLE ***

1 PLACE CE KEY IN CE MODE (IF CONTROL STORAGE IS TO BE CHANGED).
2 PLACE MODE SWITCH IN MAIN STORAGE POSITION.
3 PLACE DIAGNOSTIC CONTROL SWITCH IN LOAD STOR POSITION.
4 PLACE STARTING ADDRESS INTO SWITCHES ABCD (USE TRUE ADDRESS, DO NOT USE 0XXX FOR CONTROL ADDRESSES).
5 DEPRESS SYSTEM RESET AND CONTROL ADDRESS SET
6 DIAL SWITCHES ABCD TO DATA DESIRED IN MEMORY ADDRESS YOU CONTROL ADDRESS SET TO.
7 DEPRESS START ONCE (THIS LOADS MEMORY WITH 2 BYTES OF INFORMATION SET IN SWITCHES).
8 DIAL NEXT 2 BYTES OF DATA INTO ABCD AND REPEAT STEPS 7&8 UNTIL ALL DATA HAS BEEN LOADED.

*** HANDLOAD ROUTINE FOR 2311 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	P0=0	ZERO OUT SWITCH
0016	8076	BR	BRANCH TO LOCATION 0076
0076	2490	SET MODE K=09	SET 2311 MODE
0078	2813	I0=0$K01	SET I-REG
007A	2907	I1=0	TO 0100
007C	4A86	T=I	MOVE 0100 TO T-REGISTER
007E	2F15	H1=0$K10	
0080	2EX5	HC=0$KX0	SET UP MODULE SELECT * NOTE *
0082	2D23	P1=0$K02	SET UP RETURN BIT (BIT 6)
0084	4D9F	FF0=I1	SEND ZERO TO FILE FLAGS OUT
0086	0E08	RST FIB K=40	ISSUE INITIAL RESET
0088	0E04	RST FIB K=20	ISSUE COLD-START RESET
008A	49EF	MS=H0	LOAD MODULE SELECT REGISTER
008C	4FDF	FB0=P1	SEND RETURN TO FILE BUS OUT
008E	4BFF	TGR0=H1	MOVE CONTROL BIT TO TAG REG
0090	2E63	HC=0$K06	SET UP FILE OP, COUNT
0092	3E15	H0=H0$K10	OF ONE, DATA READ
0094	4FEF	FOP=H0	MOVE OP TO FILE OP REGISTER
0096	2E43	HC=0$K04	SET UP SELECT HEAD
0098	E598	BR IF DASI2=0	BR TO ITSELF IF NO GATED ATT.
009A	4EEF	FB0=H0	MOVE 04 TO FILE BUS OUT
009C	4BFF	TGR0=H1	MOVE CNTRL BIT TO TAG REG OUT
009E	3D00	SET FIA K=80	SET GO LATCH
00A0	E5A1	BR IF DASI2=1	BR TO ITSELF IF GATED ATTEN.
00A2	1DC0	RST FIA K=80	RESET TRAP LATCH
00A4	50EF	HC=TGRI	GET TAG REGISTER IN
00A6	CA25	BR IF H04=1	BR BACK ONE WORD IF NO TRAP
00A8	C1A9	BR IF DASI4=1	LOOP ON ITSELF IF ERROR
00AA	810A	BR TO 010A	BR TO BOOTSTRAP

* NOTE * THE X IN THE CONTROL WORD HAS THE FOLLOWING SIGNIFICANCE-

X=8 SELECT DRIVE NUMBER 0
X=4 SELECT DRIVE NUMBER 1
X=2 SELECT DRIVE NUMBER 2
X=1 SELECT DRIVE NUMBER 3

*** HANDLOAD ROUTINE FOR 2560 ***

ADDR	WORD	STATEMENT	COMMENT
0010	3210	SET MMSK K=81	BLOCK ALL TRAPS
0012	2610	SET BC K=01	SET LOGOUT LATCH
0014	2C07	P0=0	ZERO HANDLOAD FLAG REGISTER
0016	2413	G0=0$K01	BUILD HIGH HALF CS ADDR. 0100
0018	80AC	BR	BR TO 00AC
00AC	240E	SET MODE K=70	PUT IN MOD/20, 2560 MODE
00AE	2F04	SET MFA K=20	SELECT SEC FEED (NOTE 2)
00B0	2507	G1=0	LOW HALF CS ADDR. G=0100
00B2	5FDF	P1=MFD8	
00B4	DAB8	BR IF MFT5=0	BR TO 00B8 IF NPRO REQ
00B6	ED33	BR IF P12=1	BR IF NOT READY
00B8	2F10	SET MFA K=01	SET READ EX.
00BA	DAC6	BR IF MFT5=0	CHECK NPRO
00BC	CAB9	BR IF MFT4=1	BR ON NO DATA AVAIL
00BE	58FF	H1=MFR1	READ 1/2 BYTE
00C0	5AEF	H0=MFR2	READ THE OTHER 1/2
00C2	4EF3	H1=H0XH+H1L	PUT TWO 1/2 BYTES TOGETHER
00C4	6F48	STB H1 CS,G+1	STORE THEM
00C6	2B14	SET MFC K=21	RST NPRO, AND RD. EX.
00C8	055D	Z=G1<K50	CHECK FOR 80 BYTES
00CA	C4B4	BR IF ZNZ	IF NOT 80, GO TO 00B4
00CC	8100	BR	BR TO BOOTSTRAP ADDR 0100

NOTE 2 - IN ORDER TO USE THE PRIMARY FEED, REPLACE THIS WORD WITH 2F80. THE START KEY MUST BE PRESSED AT THE END TO COMPLETE CSL (LAST CARD). THIS PROCEDURE CAN ONLY BE USED FOR HANDLOADS, (BECAUSE OF CHECK SUM), AND IS NOT TO BE USED TO RUN OTHER THAN *EXX DECKS.

```
**************************
*                        *
* FOR BOOTSTRAP          *
* INFORMATION, REFER     *
* TO THE AKXXX LOGIC     *
* PAGES.                 *
**************************
```

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BCPL 001	T		CSL	RESIDENT ROUTINE	
		BCPL 002	ATABLE	ADDR=0000			
0000	8000	BCPL 003	LOCKON	003	LOCKON	BR	
0002	0BAD	BCPL 004	C			XCTL'0BAD0BAD0BAD0BAD0BAD0BAD'	
000E	80AC	BCPL 005	N2560	099	RD2560	BR	

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS

                BCPL 006    *           THE FOLLOWING WORD
                BCPL 007    *            IS THE FIRST WORD
                BCPL 008    *             OF THE CSL TRAP
0010    3210    BCPL 009                            SET MMSK K=81      BLOCK ALL TRAPS
0012    2610    BCPL 010                            SET BC K=01              SET LOG OUT LATCH
0014    50CF    BCPL 011                            P0=SWAB                  READ
0016    51AF    BCPL 012                            T0=SWCD                   SWITCHES
0018    2413    BCPL 013                            G0=0$K01                 SET UP C S ADDRESS
001A    0CCF    BCPL 014                            Z=P0<KCC           CHECK FOR
001C    C4AD    BCPL 015                  035  CHANEL  BR IF Z=0           CHANNEL
001E    0CDF    BCPL 016                            Z=P0<KDD           CHECK FOR
0020    C48F    BCPL 017                  005  N2560   BR IF Z=0           2560
0022    0CEF    BCPL 018                            Z=P0<KEE           CHECK FOR
0024    C4E3    BCPL 019                  062  N2540   BR IF Z=0           2540
0026    0CFF    BCPL 021                            Z=P0<KFF           CHECK FOR
0028    C4A8    BCPL 022    SWIERR        022  SWIERR  BR IF ZNZ           2311
002A    80DE    BCPL 023                  128  FILE    BR
002C    2486    BCPL 035    CHANEL                    SET MODE K=38      CHAN  PUT IN CHANNEL MODE
002E    2507    BCPL 036    AGAIN                     G1=0               CHAN  CS ADDRESS
0030    2B08    BCPL 037                              SET GA K=40        CHAN
0032    C9B3    BCPL 038    OPIN          038  OPIN    BR IF GT4=1        CHAN  BR ON OP IN
0034    4FAF    BCPL 039                              GB/OUT=T0          CHAN  SEND OUT DEVICE ADDRESS
0036    2B04    BCPL 040                              SET GA K=20        CHAN  RAISE ADDR.
0038    2B44    BCPL 041                              SET GA K=24        CHAN   AND SEL OUT
003A    C9BA    BCPL 042    OPINOT        042  OPINOT  BR IF GT4=0        CHAN  WAIT FOR OP IN
003C    2B40    BCPL 043                              SET GA K=04        CHAN  RESET ADDR. OUT
003E    CDBE    BCPL 044    ADDIN         044  ADDIN   BR IF GT0=0        CHAN  WAIT FOR ADDR IN
0040    2B23    BCPL 045                              T1=0$K02           CHAN  BUILD READ CMND
0042    4FBF    BCPL 046                              GB/OUT=T1          CHAN  SEND OUT THE CMND
0044    2B42    BCPL 047                              SET GA K=14        CHAN  RAISE COMMAND OUT
0046    FDC6    BCPL 048    STATUS        048  STATUS  BR IF GT3=0        CHAN  WAIT FOR STATUS
0048    5FBF    BCPL 049                              T1=GB/IN           CHAN  READ STATUS
004A    C4CA    BCPL 050    ERRORA        050  ERRORA  BR IF ZNZ           CHAN  LOOP INVALID INITIAL STATS
004C    2B48    BCPL 051    STATC                     SET GA K=44        CHAN  SET SERVICE OUT
004E    FDCF    BCPL 052    STATB         052  STATB   BR IF GT3=1        CHAN  STATUS  LOOP
0050    EDCE    BCPL 053                  052  STATB   BR IF GT2=0        CHAN  SERVICE LOOP
0052    5FFF    BCPL 054                              H1=GB/IN           CHAN  GET DATA
0054    F05B    BCPL 055                  058  FIRSTC  BR IF G07=1        CHAN  FIRST TIME  BOOTSTRAP
0056    7F48    BCPL 056                              STB H1 AS,G+1      CHAN   NOT 1ST PUT IN AUX
0058    F05C    BCPL 057                  059  DECREM  BR IF G07=0        CHAN  UNCONDIT BRANCH
005A    6F48    BCPL 058    FIRSTC                    STB H1 CS,G+1      CHAN   1ST TIME PUT IN C S
005C    055D    BCPL 059    DECREM                    Z=G1<K50           CHAN  CHECK IF ALL
005E    C4CC    BCPL 060                  051  STATC   BR IF ZNZ           CHAN   BYTES XFERED
0060    8100    BCPL 061  C                           XCTL'8100'         CHAN  GO TO BOOTSTRAP
0062    2406    BCPL 062    N2540                     SET MODE K=30      2540  PUT IN 2540 MODE
0064    2617    BCPL 063                              D0=0$K11           2540  BUILD COMPARE
0066    2783    BCPL 064    CYCLE                     D1=0$K08           2540   ADDRESS WHERE CARD
0068    3775    BCPL 065                              D1=D1$K70          2540    IMAGE IS STORED
006A    2507    BCPL 066                              G1=0               2540  ZERO REG FOR CONTROL ADDR.
006C    4066    BCPL 067                              U=D                2540  SAVE COMPARE ADDRESS
006E    5EEF    BCPL 068    DATAIN                    H0=RPS             2540  WAIT FOR
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS
0070   DE6E   BCPL 069                 068     DATAIN     BR IF H01=0          2540  DATA AVAILIBLE
0072   5B3F   BCPL 070                                    V1=RP1               2540 READ 1ST BRUSH DATA
0074   6348   BCPL 071                                    STB V1 CS,G+1        2540 -STORE IN CONTROL STORE
0076   736A   BCPL 072                                    STB V1 AS,D-1        2540 -STORE IN AUX COMPARE TBLE
0078   F0EE   BCPL 073                 068     DATAIN     BR IF LZNZ           2540 STAY IN DATA LOOP FOR 1 CD
007A   161B   BCPL 074                                    D0=D0<K01            2540 INVERT FLAG
007C   F0E7   BCPL 075                 064     CYCLE      BR IF LZ=0           2540  CHECK FOR 2ND TRAP CYCLE
007E   8100   BCPL 076  C                                  XCTL'8100'           2540 GO TO BOOTSTRAP
0080   6EE3   BCPL 077    MOD    0                        H0=H0+H0             2311  BUILD
0082   6EE3   BCPL 078    MOD    1                        H0=H0+H0             2311   MODULE
0084   6EE3   BCPL 079    MOD    2                        H0=H0+H0             2311    NUMBER
0086   0E08   BCPL 080    MOD    3                        RST FIB K=40         2311  COLD START RESET
0088   0E04   BCPL 081                                    RST FIB K=20         2311  COLD START RESET
008A   49EF   BCPL 082                                    MS=H0                2311  SELECT MODULE
008C   4EDF   BCPL 083                                    FB0=P1               2311  SET RETURN TO ZERO P1=02
008E   4BFF   BCPL 084                                    TGR0=H1              2311  SET CONTROL        H1=10
0090   2E63   BCPL 085                                    H0=0$K06             2311  SET READ
0092   3E15   BCPL 086                                    H0=H0$K10            2311
0094   4FEF   BCPL 087                                    F0P=H0               2311   DATA OP
0096   2E43   BCPL 088                                    H0=0$K04             2311  START HEAD SELECT
0098   E598   BCPL 089    WAITFL          089     WAITFL     BR IF DASI2=0                WAIT FOR END
009A   4EEF   BCPL 090                                    FB0=H0               2311  HEAD
009C   4BFF   BCPL 091                                    TGR0=H1              2311   SELECT
009E   3D00   BCPL 092                                    SET FIA K=80         2311  GO
00A0   E5A1   BCPL 093    WAIENA          093     WAIENA     BR IF DASI2=1        2311  WAIT FOR
00A2   1D00   BCPL 094                                    RST FIA K=80         2311  RESET TRAP REQ
00A4   50EF   BCPL 095    WAIENB                           H0=TGRI              2311  WAIT FOR FILE
00A6   CA25   BCPL 096                    095     WAIENB     BR IF H04=1          2311       TRAP REQUEST
00A8   C1A9   BCPL 097    ER              097     ER         BR IF DASI4=1        2311   ERROR LOOP
00AA   810A   BCPL 098  C                                  XCTL'810A'           2311    GO TO BOOTSTRAP
00AC   240F   BCPL 099    RD2560                           SET MODE K=70        2560 PUT IN 20,2560 MODE
00AE   2F04   BCPL 100                                    SET MFA K=20         2560  SELECT SEC FEED
00B0   2507   BCPL 101                                    G1=0                 2560
00B2   5FDF   BCPL 102    K5RDY                            P1=MFD8              2560  CALL IN FOR RDY CHK
00B4   DAB8   BCPL 103    K6NPO           105     FEED       BR IF MFT5=0         2560  BR IF NPRO REQ
00B6   ED33   BCPL 104                    102     K5RDY      BR IF P12=1          2560  BR IF NOT RDY
00B8   2F10   BCPL 105    FEED                             SET MFA K=01         2560  SET READ EXECUT
00BA   DAC6   BCPL 106                    112     NDIT       BR IF MFT5=0         2560  BRANCH IF NPRO
00BC   CAB9   BCPL 107                    105     FEED       BR IF MFT4=1         2560  DATA AVAIL
00BE   58FF   BCPL 108                                    H1=MFR1                   READ 1 / 2 BYTE
00C0   5AEF   BCPL 109                                    H0=MFR2                    READ OTHER HALF
00C2   4EF3   BCPL 110                                    H1=H0XH+H1L                   PUT THEM TOGETHER
00C4   6F48   BCPL 111                                    STB H1 CS,G+1        2560 STORE DATA IN C S
00C6   2B14   BCPL 112    NDIT                             SET MFC K=21         2560  RST RD EX AND NPRO
00C8   055D   BCPL 113                                    Z=G1<K50             2560
00CA   C4B4   BCPL 114                    103     K6NPO      BR IF ZNZ            2560  NOT DONE BRANCH
00CC   8100   BCPL 115  C                                  XCTL'8100'           2560 GO TO BOOTSTRAP
00CE   FFFF   BCPL 117  C                                  XCTL'FFFFFFFFFFFFFFFFFFFFFFFFFFFF'
00DC   FFFF   BCPL 118  C                                  XCTL'FFFF'
00DE   2B07   BCPL 128    FILE                             T1=0                 2311  CORRECT PARITY
00E0   6A02   BCPL 129                                    STH T DC,88          2311  SAVE MODULE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

00E2    2490    BCPL 130                                    SET MODE K=09          2311  SET FILE MODE ZONE
00E4    2813    BCPL 131                                    I0=0$K01               2311  BUILD COUNT AND
00E6    2907    BCPL 132                                    I1=0                   2311   CONTROL STORAGE
00E8    4A86    BCPL 133                                    T=I                    2311    ADDRESS
00EA    4C02    BCPL 134                                    RDH P DC,88            2311  RETRIEVE MODULE
00EC    2F15    BCPL 135                                    H1=0$K10               2311  CONSTANTS FOR
00EE    5FE9    BCPL 136                                    H0=H1                  2311   SELECTION
00F0    2D23    BCPL 137                                    P1=0$K02               2311
00F2    4D9F    BCPL 138                                    FF0=I1                 2311
00F4    8811    BCPL 139                      077   MOD    N  N=P0 BITS67            2311
00F6    54E8    BCPL 140    OVRLAY                          RDH G AS,H+2           END    OVERLAY AUX
00F8    6488    BCPL 141                                    STH G CS,I+2           END    INTO CONTROL
00FA    FA76    BCPL 142                      140   OVRLAY   BR IF H07=0            END
00FC    0608    BCPL 143                                    RST BC K=40            END   RESET CSL LIGHT
00FE    88FE    BCPL 144    C                               XCTL'88FE'             END   BRANCH TO  08FE
                BCPL 145    AEND
```

```
                                    **********************************
                                    * CROSS REFERENCE FOR CSECT BCPL *
                                    **********************************
```

BCPL 003	BCPL 003	
BCPL 005	BCPL 017	
BCPL 022	BCPL 022	
BCPL 035	BCPL 015	
BCPL 038	BCPL 038	
BCPL 042	BCPL 042	
BCPL 044	BCPL 044	
BCPL 048	BCPL 048	
BCPL 050	BCPL 050	
BCPL 051	BCPL 060	
BCPL 052	BCPL 052	BCPL 053
BCPL 058	BCPL 055	
BCPL 059	BCPL 057	
BCPL 062	BCPL 019	
BCPL 064	BCPL 075	
BCPL 068	BCPL 069	BCPL 073
BCPL 077	BCPL 139	
BCPL 089	BCPL 089	
BCPL 093	BCPL 093	
BCPL 095	BCPL 096	
BCPL 097	BCPL 097	
BCPL 099	BCPL 005	
BCPL 102	BCPL 104	
BCPL 103	BCPL 114	
BCPL 105	BCPL 103	BCPL 107
BCPL 112	BCPL 106	
BCPL 128	BCPL 023	
BCPL 140	BCPL 142	

BDIA DESCRIPTIVE TEXT

THE RESIDENT MICRODIAGNOSTIC -BDIA- IS ENTERED WHEN
1. THE SYSTEM RESET KEY IS RELEASED
   (TRAP TO ADDRESS 0240)
2. THE LOAD KEY IS RELEASED
   (TRAP TO ADDRESS 0240)
3. THE CONTROL STORAGE LOAD ROUTINE -BCPL- IS FINISHED LOADING A CSL DECK.
   (BRANCH TO LABEL * VERSON *)

THE -BDIA- ROUTINE TESTS THE CPU HARDWARE NEEDED TO PERFORM A CONTROL STORAGE LOAD OPERATION. NO I/O DEVICE OR ATTACHMENT CIRCUITS ARE TESTED.

THE TESTS PERFORMED BY THE -BDIA- ROUTINE ARE -
- BRANCHING TEST
- ALU TEST
- MODE REGISTER SET/RST TEST
- STORAGE TEST
- X LINE ADDRESSING TEST
- LOCAL STORAGE SET/RST TEST
- ALU ERROR DETECTION TEST
- STORAGE DATA, CONTROL WORD, STORAGE ADDRESS TEST
- A AND B REGISTER PARITY DETECTION TEST

ERRORS ARE INDICATED BY
1. A ONE WORD BRANCH LOOP
2. A STOP WORD
3. A BRANCH TEST STOP FAILURE

A ONE WORD BRANCH LOOP IS IDENTIFIED BY THE SYSTEM LIGHT ON AND THE MANUAL LIGHT OFF. THE CLOCK WILL RUN BUT THE MICROPROGRAM DOES NOT PROGRESS.

A STOP WORD (MOVE/ARITH-WORD TYPE 3) CAUSES THE CLOCK TO STOP, THE CLOCK STOP LIGHT IS ON, THE SYSTEM LIGHT IS OFF, THE MANUAL LIGHT IS ON, AND THE ADDRESS DISPLAYED IS THE ADDRESS OF THE CONTROL WORD FOLLOWING THE STOP WORD.

A BRANCH TEST STOP FAILURE IS CAUSED BY A BRANCH ON CONDITION WORD OR RETURN WORD FAILING TO REACH A WORD THAT SETS THE DR-REGISTER. DR BIT 7 IS SET PRIOR TO EACH OF THE BRANCH TESTS, AND THE BRANCH OR RETURN MUST POINT TO A SET OF THE DR-REGISTER. THE FAILURE IS INDICATED BY THE CLOCK OFF, THE CLOCK STOP LIGHT IS ON, SYSTEM LIGHT OFF, MANUAL LIGHT ON, AND, THE ADDRESS OF THE NEXT SEQUENTIAL WORD FOLLOWING THE WORD REACHED IN ERROR, DISPLAYED IN THE CONSOLE LIGHTS.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                BDIA 001    T                              BDIA                 BASIC DIAGNOSTIC
                BDIA 002    ATABLE  ADDR=08FE
08FE    8240    BDIA 008     VERSON          018  START     BR                   START RESIDENT DIAG.
                BDIA 012    AEND
                BDIA 013    ATABLE  ADDR=0240
                BDIA 014    *                                                    DEPRESSION OF THE SYSTEM RESET
                BDIA 015    *                                                    BUTTON OR LOAD BUTTON WILL CAUSE
                BDIA 016    *                                                    A HARDWARE TRAP TO ADDRESS 0240.
                BDIA 017    *
0240    2810    BDIA 018     START                         SET DR K=01          SET DR-7. DR 7 IS USED IN
                BDIA 019    *                                                    CHECKING THE BRANCHING FUNCTIONS
                BDIA 020    *                                                    OF THE MOD 25.  A DIAG BR LATCH
                BDIA 021    *                                                    IS SET EVERYTIME A MOD 25 BRANCH
                BDIA 022    *                                                    OR RETURN FUNCTION IS EXECUTED
                BDIA 023    *                                                    AND DR 7 IS ON.  IF THE NEXT
                BDIA 024    *                                                    MICROINSTRUCTION AFTER THE BR
                BDIA 025    *                                                    OR RETURN DOES NOT ISSUE A SET
                BDIA 026    *                                                    TO THE DR REG, A HARD STOP WILL
                BDIA 027    *                                                    OCCUR AT THE COMPLETION OF THAT
                BDIA 028    *                                                    MACHINE CYCLE.
                BDIA 029    *
0242    3210    BDIA 030                                  SET MMSK K=81        SET SYSTEM RESET PRIORITY
                BDIA 031    *                                                            BIT(MMSK-8).
                BDIA 032    *
                BDIA 033    *                                                    THE FIRST WORD OF THE TRAP SETS
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
                 BDIA 034    *                                                 THE DIAG BR LATCH TO INSURE A
                 BDIA 035    *                                                 MACHINE STOP IF AN ERRONEOUS
                 BDIA 036    *                                                 BRANCH STATEMENT IS EXECUTED
                 BDIA 037    *                                                 THE SECOND WORD OF THE TRAP SETS
                 BDIA 038    *                                                 MMSK 8 TO PREVENT FURTHER TRAPS
                 BDIA 039    *                                                 FOR THE SAME SYST RST/LOAD TRAP
                 BDIA 040    *                                                 REQUEST. AS A RESULT OF THE
                 BDIA 041    *                                                 DELAYED SETTING OF MMSK 8, THE
                 BDIA 042    *                                                 TRAP WILL OCCUR TWICE FOR EACH
                 BDIA 043    *                                                 SYSTEM RST/LOAD TRAP REQUEST.
                 BDIA 044    *                                                 THEREFORE, THE FIRST TWO WORDS
                 BDIA 045    *                                                 WILL BE EXECUTED TWICE BEFORE
                 BDIA 046    *                                                 THE FOLLOWING WORD IS EXECUTED.
                 BDIA 047    *
0244     2400    BDIA 048                          SET MODE K=00
                 BDIA 049    *                                                                STORAGE ZONE 4.
                 BDIA 050    *                 ***************************************************************
                 BDIA 051    *                      BRANCHING TEST.
                 BDIA 052    *                 ***************************************************************
0246     F8CB    BDIA 053                    056  BCDR7     BR IF DR BIT7=1    BR TO 024A IF DR-7 IS ON
0248     B7C9    BDIA 054                    122  DR7OFF N  N=BAH              DR 7 SHOULD BE ON.  BRANCH ON
                 BDIA 055    *                                                   BA HIGH FOR BETTER RESOLUTION.
024A     2810    BDIA 056     BCDR7                SET DR K=01                 SET DR-7 TO CONTINUE BR TESTS.
024C     25F3    BDIA 057                          G1=0$KOF                    ATTEMPT TO SET G1 REG = 00001111
                 BDIA 058    *                                                   IF A STOP OCCURS AT THE ADDRESS
                 BDIA 059    *                                                   OF THE NEXT WORD, G1 WAS SET
                 BDIA 060    *                                                   WRONG OR THE BRANCH TEST FAILED.
024E     C54F    BDIA 061     BCG10N         061  BCG10N    BR IF G1 BIT0=1    G10=0; SHOULD NOT BRANCH.
0250     2810    BDIA 062                          SET DR K=01                 SET DR-7 TO CONTINUE BR TESTS.
                 BDIA 063    *                                                   IF A STOP OCCURS AT THE ADDRESS
                 BDIA 064    *                                                   OF THE NEXT WORD, G1 WAS SET
                 BDIA 065    *                                                   WRONG OR THE BRANCH TEST FAILED.
0252     D553    BDIA 066     BCG11N         066  BCG11N    BR IF G1 BIT1=1    G11=0; SHOULD NOT BRANCH
0254     2810    BDIA 067                          SET DR K=01                 SET DR-7 TO CONTINUE BR TESTS.
                 BDIA 068    *                                                   IF A STOP OCCURS AT THE NEXT
                 BDIA 069    *                                                   SEQUENTIAL WORD FOLLOWING THIS
                 BDIA 070    *                                                   BRANCH, G1 WAS SET WRONG OR THE
                 BDIA 071    *                                                   BRANCH TEST FAILED.
0256     F53C    BDIA 072                    075  BCG12Y    BR IF G1 BIT2=0    G12=0; SHOULD BRANCH TO ADR 023C
                 BDIA 073    AEND
                 BDIA 074    ATABLE  ADDR=023C
023C     2810    BDIA 075     BCG12Y                SET DR K=01                 SET DR-7 TO CONTINUE BR TESTS.
                 BDIA 076    *                                                   IF A STOP OCCURS AT THE NEXT
                 BDIA 077    *                                                   SEQUENTIAL WORD FOLLOWING THIS
                 BDIA 078    *                                                   BRANCH, G1 WAS SET WRONG OR THE
                 BDIA 079    *                                                   BRANCH TEST FAILED.
023E     F56C    BDIA 080                    083  BCG13Y    BR IF G1 BIT3=0    G13=0; SHOULD BRANCH TO ADR 026C
                 BDIA 081    AEND
                 BDIA 082    ATABLE  ADDR=026C
026C     2810    BDIA 083     BCG13Y                SET DR K=01                 SET DR-7 TO CONTINUE BR TESTS.
026E     9FFC    BDIA 084                    087  UB1FFC    BR                 UNCONDITIONAL BRANCH TO ADR 1FFC
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                BDIA 085    AEND
                BDIA 086    ATABLE  ADDR=1FFC
1FFC   2810     BDIA 087     UB1FFC                        SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
1FFE   D57F     BDIA 088    HCTRL                          BR IF G1 BIT1=1    NO BRANCH
                BDIA 089    AEND
                BDIA 090    ATABLE  ADDR=2000
2000   2810     BDIA 091                                   SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
2002   25F5     BDIA 092                                   G1=0$KF0           ATTEMPT TO SET G1 REG = 11110000
                BDIA 093    *                                                  IF A STOP OCCURS AT THE ADDRESS
                BDIA 094    *                                                  OF THE NEXT WORD, G1 WAS SET
                BDIA 095    *                                                  WRONG OR THE BRANCH TEST FAILED.
2004   E504     BDIA 096     BCG12N          096  BCG12N    BR IF G1 BIT2=0    G12=1; SHOULD NOT BRANCH
2006   2810     BDIA 097                                   SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
                BDIA 098    *                                                  IF A STOP OCCURS AT THE ADDRESS
                BDIA 099    *                                                  OF THE NEXT WORD, G1 WAS SET
                BDIA 100    *                                                  WRONG OR THE BRANCH TEST FAILED.
2008   F508     BDIA 101     BCG13N          101  BCG13N    BR IF G1 BIT3=0    G13=1; SHOULD NOT BRANCH
200A   2810     BDIA 102                                   SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
200C   A040     BDIA 103                        106  UB2040    BR             UNCONDITIONAL BRANCH TO ADR 2040
                BDIA 104    AEND
                BDIA 105    ATABLE  ADDR=2040
2040   2810     BDIA 106     UB2040                        SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
                BDIA 107    *                                                  IF A STOP OCCURS AT THE NEXT
                BDIA 108    *                                                  SEQUENTIAL WORD FOLLOWING THIS
                BDIA 109    *                                                  BRANCH, G1 WAS SET WRONG OR THE
                BDIA 110    *                                                  BRANCH TEST FAILED.
2042   C55F     BDIA 111                        114  BCG10Y    BR IF G1 BIT0=1    G10=1; SHOULD BRANCH TO ADR 205E
                BDIA 112    AEND
                BDIA 113    ATABLE  ADDR=205E
205E   2810     BDIA 114     BCG10Y                        SET DR K=01        SET DR-7 TO CONTINUE BR TESTS.
                BDIA 115    *                                                  IF A STOP OCCURS AT THE NEXT
                BDIA 116    *                                                  SEQUENTIAL WORD FOLLOWING THIS
                BDIA 117    *                                                  BRANCH, G1 WAS SET WRONG OR THE
                BDIA 118    *                                                  BRANCH TEST FAILED.
2060   D57F     BDIA 119                        125  BCG11Y    BR IF G1 BIT1=1    G11=1; SHOULD BRANCH TO ADR 207E
                BDIA 120    AEND
                BDIA 121    ATABLE  ADDR=0440
0440   5007     BDIA 122     DR7OFF 0                      STOP               CPU MODE AND DR 7 OFF.
                BDIA 123    AEND
                BDIA 124    ATABLE  ADDR=207E
207E   2800     BDIA 125     BCG11Y                        SET DR K=00        THIS COMPLETES BDIA BRANCHING
                BDIA 126    *                                                                 TESTS.
                BDIA 127    *              ***************************************************************
                BDIA 128    *                 ALU TEST -MANY OF THE WORDS IN THIS TEST CAUSE ALU ERRORS IF THE CKT
                BDIA 129    *                 BEING TESTED FAILS AND THEREFORE DO NOT HAVE TO BE TESTED BY BR WDS
                BDIA 130    *              ***************************************************************
2080   3E09     BDIA 131                                   H0=0-K00       **FF=00-00 WITH NO CARRY
2082   C483     BDIA 132     DC0ER             132  DC0ER     BR IF Z=0          CHK FOR DYN BIT 0 BEING OFF.
2084   F085     BDIA 133     DC7ER             133  DC7ER     BR IF LZ=0          CHK FOR DYN BIT 7 BEING OFF.
2086   E087     BDIA 134     DC6ER             134  DC6ER     BR IF HZ=0          CHK FOR DYN BIT 6 BEING OFF.
2088   F489     BDIA 135     ACONER            135  ACONER    BR IF AC=1         CHK FOR DYN BIT 3 BEING OFF
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT        COMMENTS
208A  6EE5    BDIA 136                                  H0=H0$H0      **FF=FF$FF
208C  A0DA    BDIA 137                 195   CK4FF      BAL           **BRANCH TO CHECK H0 FOR FF
208E  2E0D    BDIA 138                                  H0=H0+K00     **FF=FF+00
              BDIA 139  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 140  *                                              OF THE WORD BELOW, THE CHECK FOR
              BDIA 141  *                                              FF HAS FAILED.
2090  A0DA    BDIA 142                 195   CK4FF      BAL
2092  7EE1    BDIA 143                                  H0=H0-H0+1    **00=FF-FF+1 WITH A CARRY
              BDIA 144  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 145  *                                              OF THE WORD BELOW, THE CHECK FOR
              BDIA 146  *                                              FF HAS FAILED.
2094  F494    BDIA 147   NOACER        147   NOACER     BR IF AC=0    **SHOULD HAVE DYN 3 BIT ON
2096  7EE1    BDIA 148                                  H0=H0-H0+1    **00=00-00+1
2098  6EE3    BDIA 149                                  H0=H0+H0      **00=00+00
209A  2E5F    BDIA 150                                  H0=H0+K55     **55=00+55
209C  6EE3    BDIA 151                                  H0=H0+H0      **AA=55+55
209E  6EE9    BDIA 152                                  H0C=H0+H0+1   **55=AA+AA+1
20A0  0E5F    BDIA 153                                  Z=H0<K55      **00=55<55
20A2  C4A2    BDIA 154   WT3ER1        154   WT3ER1     BR IF ZNZ     **H0 SHOULD BE 55 AND DYN BIT0=1
20A4  2E87    BDIA 155                                  H0=0$K88      **88=00$88
20A6  3E27    BDIA 156                                  H0=H0$K22     **AA=88$22
20A8  2EEF    BDIA 157                                  H0=H0+KEE     **98=AA+EE
20AA  2E77    BDIA 158                                  H0=0$K77      **77=00$77
20AC  1E87    BDIA 159                                  H0=H0*-K88    **77=77*-88
20AE  1EB7    BDIA 160                                  H0=H0*-KBB    **44=77*-BB
20B0  2EF7    BDIA 161                                  H0=0$KFF      **FF=00$FF
20B2  0E11    BDIA 162                                  Z=H0+K01      **00=FF+01
20B4  F0B4    BDIA 163    DC7ERR       163   DC7ERR     BR IF LZNZ             CHK FOR DYN BIT 7 BEING ON.
20B6  E0B6    BDIA 164    DC6ERR       164   DC6ERR     BR IF HZNZ             CHK FOR DYN BIT 6 BEING ON.
20B8  1EFB    BDIA 165                                  H0=H0<K0F     **F0=FF<0F
20BA  A0E2    BDIA 166                 201   CK4F0      BAL           **BR TO CHECK H0 FOR F0
20BC  1EF3    BDIA 167                                  H0=H0*-K0F    **F0=F0*-0F
              BDIA 168  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 169  *                                              OF THE WORD BELOW, THE CHECK FOR
              BDIA 170  *                                              F0 HAS FAILED.
20BE  A0E2    BDIA 171                 201   CK4F0      BAL           **BR TO CHECK H0 FOR F0
20C0  3E15    BDIA 172                                  H0=H0$K10     **F0=F0$10
              BDIA 173  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 174  *                                              OF THE WORD BELOW, THE CHECK FOR
              BDIA 175  *                                              F0 HAS FAILED.
20C2  A0E2    BDIA 176                 201   CK4F0      BAL           **BR TO CHECK H0 FOR F0
20C4  3E0D    BDIA 177                                  H0=H0-K00     **EF=F0-00
              BDIA 178  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 179  *                                              OF THE WORD BELOW, THE CHECK FOR
              BDIA 180  *                                              F0 HAS FAILED.
20C6  2E1D    BDIA 181                                  H0=H0+K10     **FF=EF+10
20C8  2FF5    BDIA 182                                  H1=0$KF0      **F0=00+F0
20CA  A0DA    BDIA 183                 195   CK4FF      BAL                **BR TO CHECK H0 FOR FF
20CC  0FFD    BDIA 184                                  Z=H1<KF0      **00=F0<F0
              BDIA 185  *                                              IF A STOP OCCURS AT THE ADDRESS
              BDIA 186  *                                              OF THE WORD BELOW, THE CHECK FOR
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
                BDIA 187    *                                                 F0 HAS FAILED.
20CE    C4CE    BDIA 188     ALUER2      188    ALUER2     BR IF ZNZ         **H1 SHOULD BE F0 AND DYN BIT 0=1
20D0    1FF5    BDIA 189                                    H1=H1*-KF0        **00=F0*-F0
20D2    C4D2    BDIA 190     ALUER3      190    ALUER3     BR IF ZNZ         **H1 SHOULD BE ZERO AND DYN BIT 0=1
20D4    2EF5    BDIA 191                                    H0=0$KF0          **F0=00$F0
20D6    A0E2    BDIA 192                 201    CK4F0      BAL               **BR TO CHECK H0 FOR F0
20D8    A0E8    BDIA 193                 207    MORTST     BR                **BR TO MODE REG SET/RESET TEST
                BDIA 194    *           ********************************************************************
20DA    0EFF    BDIA 195     CK4FF                          Z=H0<KFF                HO SHOULD BE FF
20DC    C4E1    BDIA 196                 198    ZISOK      BR IF Z=0                       AND DYN BIT 0=1.
20DE    2810    BDIA 197     ANOTF0                         SET DR K=01             SET DR 7 AND DO A RTN WHICH
20E0    128E    BDIA 198     ZISOK                          RTN                           WILL CAUSE A STOP WITH THE
                BDIA 199    *                                                             ADDRESS OF THE BAL TO
                BDIA 200    *                                                             THIS SUBROUTINE + 4.
20E2    0EFD    BDIA 201     CK4F0                          Z=H0<KF0                HO SHOULD BE F0
20E4    C4DE    BDIA 202                 197    ANOTF0     BR IF ZNZ                       AND DYN BIT 0=1.
20E6    128E    BDIA 203                                    RTN
                BDIA 204    *           ********************************************************************
                BDIA 205    *             TEST MODE REG FOR SET/RESET
                BDIA 206    *           ********************************************************************
20E8    24F6    BDIA 207     MORTST                         SET MODE K=3F **SET LS AND EXT MODE TO 7
20EA    578F    BDIA 208                                    I0=BA           ** I0=XIIIXIII
20EC    1887    BDIA 209                                    I0=I0*-K88      ** I0=77
20EE    087F    BDIA 210                                    Z=I0<K77
20F0    E0F0    BDIA 211     MDSTHI      211    MDSTHI     BR IF HZNZ              LOOP IF MODE REG 2-4 NOT ALL ON
20F2    F0F2    BDIA 212     MDSTLO      212    MDSTLO     BR IF LZNZ              LOOP IF MODE REG 5-7 NOT ALL ON
20F4    2400    BDIA 213                                    SET MODE K=00 **SET LS AND EXT MODE TO ZERO
20F6    574F    BDIA 214                                    G0=BA           **G0=X000X000
20F8    0487    BDIA 215                                    Z=G0*-K88       **Z=00
20FA    E0FA    BDIA 216     MDRTHI      216    MDRTHI     BR IF HZNZ              LOOP IF MODE REG 2-4 NOT ALL OFF
20FC    F0FC    BDIA 217     MDRTLO      217    MDRTLO     BR IF LZNZ              LOOP IF MODE REG 5-7 NOT ALL OFF
                BDIA 218    *           ********************************************************************
                BDIA 219    *             BEGIN STORAGE TEST
                BDIA 220    *           ********************************************************************
20FE    2EF7    BDIA 221                                    H0=0$KFF
2100    2FF7    BDIA 222                                    H1=0$KFF        **H=FFFF
2102    6E02    BDIA 223                                    STH H DC,88     **CTRL ADDR 0388=FFFF
2104    2E05    BDIA 224                                    H0=0$K00
2106    2F05    BDIA 225                                    H1=0$K00        **H=0000
2108    4E02    BDIA 226                                    RDH H DC,88             **READ CTRL STOR 0088 INTO H REG
210A    0FFF    BDIA 227                                    Z=H1<KFF        **00=FF<FF
210C    E08C    BDIA 228     DCR1ER      228    DCR1ER     BR IF HZNZ              LOOP IF H1 HIGH IS NOT F
210E    F08E    BDIA 229     DCR2ER      229    DCR2ER     BR IF LZNZ              LOOP IF H1 LOW IS NOT F
2110    1EFF    BDIA 230                                    H0=H0<KFF       **00=FF<FF   H=00FF
2112    C492    BDIA 231     DCR0ER      231    DCR0ER     BR IF ZNZ       **SHOULD HAVE READ FF IN TO H0
2114    2E33    BDIA 232                                    H0=0$K03
2116    2F87    BDIA 233                                    H1=0$K88        ** H=0388
2118    6FE8    BDIA 234                                    STB H1 CS,H+1 ** CTRL ADDR 0388=88FF  H=0389
211A    6FEA    BDIA 235                                    STB H1 CS,H-1 ** CTRL ADDR 0388=8889  H=0388
211C    44E8    BDIA 236                                    RDH G CS,H+2  ** G=8889 H=038A
211E    2F13    BDIA 237                                    H1=0$K01        ** H=0301
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
2120  C48F    BDIA 238                                  Z=G0<K88       **00=88<88
2122  E0A2    BDIA 239    DCR3ER       239  DCR3ER     BR IF HZNZ           LOOP IF H1 HIGH IS NOT 8
2124  F0A4    BDIA 240    DCR4ER       240  DCR4ER     BR IF LZNZ           LOOP IF H1 LOW IS NOT 8
2126  257F    BDIA 241                                  G1=G1+K77      **00=89+77
2128  C4A8    BDIA 242    CSR1ER       242  CSR1ER     BR IF ZNZ      **SHOULD HAVE READ 89 IN TO G1
212A  2E05    BDIA 243                                  H0=0$K00       ** H=0001
212C  55EA    BDIA 244                                  RDB G1 AS,H-1  **H=0000
212E  C4AE    BDIA 245    BMODER       245  BMODER     BR IF ZNZ      **ARITH MODIFIER OFF B REG FAILED IF BR
2130  2E45    BDIA 246                                  H0=0$K40       ** H=4000
2132  3FE9    BDIA 247                                  H1=0-KE0       ** H=401F
2134  25F7    BDIA 248    AUXSTT                        G1=0$KFF       **G=88FF
2136  75E0    BDIA 249                                  STB G1 AS,H    **1ST PASS AUX ADDR 401E=XXFF H=401F
              BDIA 250  *                                               **2ND PASS AUX ADDR 401E=FFFF H=401E
2138  2505    BDIA 251                                  G1=0$K00       **G=8800
213A  55EA    BDIA 252                                  RDB G1 AS,H-1  **G=88FF AFTER 1ST PASS, H=401E
              BDIA 253  *                                               **       AFTER 2ND PASS, H=401D
213C  05FF    BDIA 254                                  Z=G1<KFF       **00=FF<FF
213E  C4BE    BDIA 255    AUXERR       255  AUXERR     BR IF ZNZ      **SHOULD HAVE READ FF IN TO G1
2140  FB34    BDIA 256                  248  AUXSTT     BR IF H1 BIT7=0
2142  44E6    BDIA 257                                  G=H            ** G=401D H=401D  DBL BYTE MOD TEST
2144  044D    BDIA 258                                  Z=G0<K40       ** 00=40<40
2146  C4C6    BDIA 259    DBMER0       259  DBMER0     BR IF ZNZ      ** SHOULD HAVE MOVED 40 FROM H0 TO G0.
2148  75F1    BDIA 260                                  G1=G1-H1+1     ** 00=1D-1D+1
214A  C4CA    BDIA 261    DBMER1       261  DBMER1     BR IF ZNZ      ** SHOULD HAVE MOVED 1D FROM H1 TO G1
214C  C060    BDIA 262                                  RST S K=06           RESET S5 AND S6.
214E  88EC    BDIA 263                  269  LSADDR     BR             **BR TO X LINE ADDRESSING TEST
              BDIA 264  AEND
              BDIA 265  *          ************************************************************
              BDIA 266  *                          X LINE ADDRESSING
              BDIA 267  *          ************************************************************
              BDIA 268  ATABLE  ADDR=08EC
08EC  D1ED    BDIA 269   LSADDR          269  LSADDR     BR IF S5=1           NO BR. CHECK S5 OFF.
08EE  E1EF    BDIA 270   CKS60F          270  CKS60F     BR IF S6=1           NO BR. CHECK S6 OFF.
08F0  2E87    BDIA 271                                  H0=0$K88             XXXK ADDRESSABLE 8A AND 8C
08F2  2F05    BDIA 272                                  H1=C$KC0             X    SET TO 8800 WHICH IS
08F4  6E12    BDIA 273                                  STH H DC,8A          X     BR TO 0800
08F6  6E22    BDIA 274                                  STH H DC,8C          X
08F8  3EE9    BDIA 275                                  H0=0-KE0             0000 H SET TO 1F77. THIS IS THE
08FA  2F77    BDIA 276                                  H1=0$K77             0    INIT VALE REQ'D FOR A=0$KK
08FC  8802    BDIA 277                  281  BEGMOD     BR                   BR TO 0802
              BDIA 278  AEND
              BDIA 279  ATABLE  ADDR=0800
0800  4E02    BDIA 280   MODCWD                         RDH H DC,88          XXXX MODIFY CTRL WD AT K88.
0802  2E1B    BDIA 281   BEGMOD                         H0=H0+K01            X
0804  2F1D    BDIA 282                                  H1=H1+K10            X
0806  CA0D    BDIA 283                  302  CKPASS     BR IF H0 BIT4=1      X BR AFTER 8TH PASS..
0808  6E02    BDIA 284                                  STH H DC,88          X
080A  8388    BDIA 285  C                               XCTL'8388'  BR TO K ADDR 88 AT ADDR 0388.
              BDIA 286  *              FUNCTION PERFORMED BY CTRL WORDS IN K ADDR. CTRL STORAGE
              BDIA 287  *                   FIRST PASS       SECOND PASS      THIRD PASS          FOURTH PASS
              BDIA 288  *                88 2XX7 A=0$KK   88 0XXF Z=A<KK  88 4XX3 B=AXH+BL   88 0XXF Z=A<KK
```

```
ADDR   WORD  SEQUENCE NO. LABEL     NEXTSEQ   NEXTLABEL  STATEMENT             COMMENTS

             BDIA 289   *                  ----------     -----------     ------------     -----------
             BDIA 290   *                  2087 U0=0$K88  008F Z=U0<88    4003 U0=U0XH+U0L 008F Z=U0<88
             BDIA 291   *                  2197 U1=0$K99  019F Z=U1<99    4113 U1=U1XH+U11 019F Z=U1<99
             BDIA 292   *                  22A7 V0=0$KAA  02AF Z=V0<AA    4223 V0=V0XH+V0L 02AF Z=V0<AA
             BDIA 293   *                  23B7 V1=0$KBB  03BF Z=V1<BB    4333 V1=V1XH+V1L C3BF Z=V1<BB
             BDIA 294   *                  24C7 G0=0$KCC  04CF Z=G0<CC    4443 G0=G0XH+G0L C4CF Z=G0<CC
             BDIA 295   *                  25D7 G1=0$KDD  05DF Z=G1<DD    4553 G1=G1XH+G1L C5DF Z=G1<DD
             BDIA 296   *                  26E7 D0=0$KEE  06EF Z=D0<EE    4663 D0=D0XH+D0L 06EF Z=D0<EE
             BDIA 297   *                  27F7 D1=0$KFF  07FF Z=D1<FF    4773 D1=D1XH+D1L 07FF Z=D1<FF
             BDIA 298   *               8A BR          8A BR IF ZNZ    8A BR            8A BR IF ZNZ
             BDIA 299   *                  TO MODCWD.     TO 008A.        TO MODCWD.       TO 008A.
             BDIA 300   *               8C BR          8C BR           8C BR            8C BR
             BDIA 301   *                  TO MODCWD.     TO MODCWD       TO MODCWD        TO MODCWD
080C   E1A3  BDIA 302    CKPASS       313   PASS24     BR IF S6=1            CHECK PASS. BR AFTER 2ND OR 4TH.
080E   2020  BDIA 303                                  SET S6                S6=1 TO SAY NEXT PASS IS 2 OR 4.
0810   E190  BDIA 304    CKS6E1       304   CKS6E1     BR IF S6=0            CHECK SET OF S6
0812   2EC5  BDIA 305                                  H0=0$KC0              XXXX SET 8A TO C48A WHICH IS
0814   3E43  BDIA 306                                  H0=H0$K04             X    BR IF ZNZ TO 008A
0816   2F85  BDIA 307                                  H1=0$K80              X
0818   3FA3  BDIA 308                                  H1=H1$K0A             X
081A   6E12  BDIA 309                                  STH H DC,8A           X
081C   2EF7  BDIA 310                                  H0=0$KFF              0000 H SET TO FF7F. THIS IS THE
081E   3F89  BDIA 311                                  H1=0-K80              0    INIT VALUE REQ'D FOR Z=A<KK
0820   8802  BDIA 312                 281   BEGMOD     BR                    BR TO BEGIN MODIFIC OF CTRL WD.
0822   0020  BDIA 313    PASS24                        RST S6                PASS 2 OR 4 JUST FINISHED. RESET
0824   E1A5  BDIA 314    CKS6E0       314   CKS6E0     BR IF S6=1            S6 TO INDICATE NEXT PASS IS 3RD.
0826   D1AC  BDIA 315                 318   NOT4TH     BR IF S5=0            S5=1 INDICATES 4TH PASS COMP.
0828   0040  BDIA 316                                  RST S5
082A   83D8  BDIA 317                 331   LSSET      BR                    BR TO LOCAL STOR SET/RESET TEST
082C   2040  BDIA 318    NOT4TH                        SET S5                NEXT PASS IS THIRD
082E   D1AE  BDIA 319    CKS5E1       319   CKS5E1     BR IF S5=0            CK SET OF S5.
0830   4E22  BDIA 320                                  RDH H DC,8C           XXXX RESTORE 8A TO BR TO 0800
0832   6E12  BDIA 321                                  STH H DC,8A           X
0834   3EC9  BDIA 322                                  H0=0-KC0              0000 H SET TO 3FF3. THIS IS INIT
0836   5EF1  BDIA 323                                  H1=H0X                0    VALUE REQ'D FOR B=AXH+BL
0838   8802  BDIA 324                 281   BEGMOD     BR                    BR TO BEGIN MODIFIC OF CTRL WD.
             BDIA 325   AEND
             BDIA 326   *
             BDIA 327   *         ********************************************************************
             BDIA 328   *         *        LOCAL STORAGE SET/RESET ROUTINE..... ZONE ZERO
             BDIA 329   *         ********************************************************************
             BDIA 330   ATABLE  ADDR=03D8
03D8   4812  BDIA 331    LSSET                         RDH I DC,8A           I=C48A XXX K-ADDR BC IS SET TO
03DA   3935  BDIA 332                                  I1=I1$K30             I=C4BA X   C4BC WHICH IS A
03DC   292B  BDIA 333                                  I1=I1+K02             I=C4BC X       BR IF ZNZ
03DE   68E2  BDIA 334                                  STH I DC,BC                  X   TO ITSELF.
03E0   2A25  BDIA 335                                  T0=0$K20              XXXX T SET TO 20F7 WHICH IS
03E2   2BF5  BDIA 336                                  T1=0$KF0              X        U0=0$KFF
03E4   3B73  BDIA 337                                  T1=T1$K07             X
03E6   2C25  BDIA 338                                  P0=0$K20              0000  P SET TO 201B WHICH IS
03E8   2D15  BDIA 339                                  P1=0$K10              0        U0=U0+K01
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
03EA  3DB3    BDIA 340                                   P1=P1$K0B          0
03EC  2E05    BDIA 341                                   H0=0$K00           XXXX  H SET TO 0001 WHICH IS
03EE  2F13    BDIA 342                                   H1=0$K01           X         Z=U0+K00
03F0  83D0    BDIA 343                  355  STNWCW     BR                  BR TO STORE THE THREE CONTROL
              BDIA 344   *                                                  WORDS JUST BUILT IN T, P, AND H.
              BDIA 345   AEND
              BDIA 346   ATABLE  ADDR=03C0
03C0  C4C0    BDIA 347    NOTRST          347  NOTRST     BR IF ZNZ           CK THAT ALL BITS WERE RESET.
03C2  4AC2    BDIA 348    RDOLCW                         RDH T DC,B8        READ OLD CTRL WORDS.
03C4  4CD2    BDIA 349                                   RDH P DC,BA        READ OLD CTRL WORDS.
03C6  4EF2    BDIA 350                                   RDH H DC,BE        READ OLD CTRL WORDS.
03C8  2A1B    BDIA 351                                   T0=T0+K01          MODIFY CTRL WDS...INCREMENT X.
03CA  2C1B    BDIA 352                                   P0=P0+K01          MODIFY CTRL WDS...INCREMENT X.
03CC  2E1B    BDIA 353                                   H0=H0+K01          MODIFY CTRL WDS...INCREMENT X.
03CE  E0F2    BDIA 354                  373  DYNCRT     BR IF HZNZ         BR IF SIXTEENTH PASS COMPLETED.
03D0  6AC2    BDIA 355    STNWCW                         STH T DC,B8        STORE NEW CONTROL WORDS.
03D2  6CD2    BDIA 356                                   STH P DC,BA        STORE NEW CONTROL WORDS.
03D4  6EF2    BDIA 357                                   STH H DC,BE        STORE NEW CONTROL WORDS.
03D6  83B8    BDIA 358   C                               XCTL'83B8'   BR TO K-ADDRESSABLE B8.
              BDIA 359   AEND
              BDIA 360   *          ************************************************************************
              BDIA 361   *          * THE 4 CONTROL WORDS BELOW WERE BUILT IN K-ADDRESSABLE CONTROL
              BDIA 362   *          * STORAGE AND BRANCHED TO FROM THE WORD IMMEDIATLY ABOVE THESE COMENTS.
              BDIA 363   *          * THIS IS DONE 16 DIFFERENT TIMES, WITH 'X' INCREMENTED EACH TIME TO
              BDIA 364   *          * MODIFY THE FUNCTION OF THE WORDS.
              BDIA 365   *          * K-ADDR B8...CTL STOR ADDR 03B8 EQUALS   2XF7   A=A$KFF
              BDIA 366   *          * K-ADDR BA...CTL STOR ADDR 03BA EQUALS   2X1B   A=A+K01
              BDIA 367   *          * K-ADDR BC...CTL STOR ADDR 03BC EQUALS   C4BC   BR IF ZNZ TO ITSELF
              BDIA 368   *          * K-ADDR BE...CTL STOR ADDR 03BE EQUALS   0X01   Z=A+K00
              BDIA 369   *          *
              BDIA 370   *          ************************************************************************
              BDIA 371   ATABLE  ADDR=03F2
              BDIA 372   *          ************************************************************************
03F2  2505    BDIA 373    DYNCRT                         G1=0$K00       **SET G1 TO ZERO AND THEN INCREMENT TO
03F4  251B    BDIA 374    ZINCR                          G1=G1+K01      ** OBTAIN ALL BIT COMBINATIONS ON Z BUS
03F6  F4FD    BDIA 375                  381  ALUEDT     BR IF AC=1     **END OF TEST--BR TO ALU OE TEST
03F8  C4F4    BDIA 376                  374  ZINCR      BR IF ZNZ      **BR BACK TO CHECK ALL Z BUSS BIT COMB
03FA  5007    BDIA 377                                   STOP           **DYN COND REG BIT 0 IS ON IN ERROR
              BDIA 378   *          ************************************************************************
              BDIA 379   *             ALU ERROR DECTECTION TEST
              BDIA 380   *          ************************************************************************
03FC  2413    BDIA 381    ALUEDT                         G0=0$K01
03FE  883C    BDIA 382                  386  ZEROG1     BR
              BDIA 383   AEND
              BDIA 384   ATABLE  ADDR=083A
083A  6443    BDIA 385    CKALOE                         G0=G0+G0
083C  2505    BDIA 386    ZEROG1                         G1=0$K00
083E  7543    BDIA 387                                   G1=G1-G0
0840  3808    BDIA 388                                   SET DR K=C0    **DISABLE STOP ON ERROR & FORCE PLUS
0842  050D    BDIA 389                                   Z=G1<K00       ** SIDE OF ALU A ENTRY CAUSING ALU CK
0844  3800    BDIA 390                                   SET DR K=80
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
0846  DAC6  BDIA 391      ALCKER     391    ALCKER     BR IF MC BIT5=0 **ALU ERROR SHOULD HAVE SET MC 5
0848  1212  BDIA 392                                   RST MMSK K=91
084A  2800  BDIA 393                                   SET DR K=00
084C  C43A  BDIA 394                 385    CKALOE     BR IF GO BIT0=0 **BRANCH BACK TO CHECK EACH OE CKT
            BDIA 395    *          ***************************************************************
            BDIA 396    *            STORAGE DATA,CONTROL WORD,STORAGE ADDRESS, A REG AND B REG
            BDIA 397    *            PARITY DETECTION TEST----DC REG BITS 3,6 AND 7 MUST BE OFF
            BDIA 398    *          ***************************************************************
084E  2440  BDIA 399                                   SET MODE K=04
0850  886E  BDIA 400                 415    CKSTPC     BAL                **STORE ADDR OF CHECK DATA TO BE READ
0852  0E3F  BDIA 401                                   Z=H0<K33           **THIS ADDR CONTAINS 0E3F
0854  FED4  BDIA 402      ROCRCE     402    ROCRCE     BR IF MC3=0        **SHOULD HAVE R0 CTRL WORD PARITY CHECK
0856  4E80  BDIA 403                                   RDH H CS,I         **H=0E3F WITH EVEN PARITY IN H0
0858  6E80  BDIA 404                                   STH H CS,I         **RESTORE CHECK DATA WITH GOOD PARITY
085A  FADA  BDIA 405      NOBPC      405    NOBPC      BR IF MC7=0        **SHOULD HAVE SET B REG PARITY CK LATCH
085C  0E3F  BDIA 406                                   Z=H0<K33           **SET A REG PARITY CK LT & RST DC3,6&7
085E  EADE  BDIA 407      NOAPC      407    NOAPC      BR IF MC6=0        **SHOULD HAVE SET A REG PARITY CK LATCH
0860  1212  BDIA 408                                   RST MMSK K=91      **RESET MC REG
0862  886E  BDIA 409                 415    CKSTPC     BAL                **STORE ADDR OF CHECK DATA TO BE READ
0864  0F1F  BDIA 410                                   Z=H1<K11           **THIS ADDR CONTAINS 0F1F
0866  FEE6  BDIA 411      R1CRCE     411    R1CRCE     BR IF MC3=0        **SHOULD HAVE R1 CTRL WORD PARITY CHECK
0868  4E80  BDIA 412                                   RDH H CS,I         **H=0F1F WITH EVEN PARITY IN H1
086A  6E80  BDIA 413                                   STH H CS,I         **RESTORE CHECK DATA WITH GOOD PARITY
086C  C2E2  BDIA 414                                   RTN H MMSK1=0      **CAUSE STOR ADDR CK--H1=1F WITH P BIT
086E  4E80  BDIA 415      CKSTPC                       RDH H CS,I         **1ST LOOP H=0E3F,ON 2ND LOOP H=0F1F
0870  3804  BDIA 416                                   SET DR K=A0        **DISABLE STOP ON ERROR AND FORCE
0872  6E80  BDIA 417                                   STH H CS,I         **  STORE BITS P0 AND P1 TO MEMORY
0874  CAF4  BDIA 418      NOSDPC     418    NOSDPC     BR IF MC4=0        **SHOULD HAVE SET STOR DATA PARITY LT
0876  3800  BDIA 419                                   SET DR K=80        **RST FORCING OF STORE BITS P0 AHD P1
0878  128E  BDIA 420                                   RTN                **RTN FOR ADDITIONAL TESTING & 2ND LOOP
            BDIA 421    AEND
            BDIA 422    ATABLE  ADDR=0F1E
0F1E  EE9E  BDIA 423      NOSTAC     423    NOSTAC     BR IF MC2=0        **SHOULD HAVE SET STORAGE ADDRESS CHECK
0F20  1212  BDIA 424                                   RST MMSK K=91      **RESET MC REG
0F22  5EFF  BDIA 425                                   H1=MC              **CHECK RESET OF MC REG
0F24  C4A4  BDIA 426      MCRSTE     426    MCRSTE     BR IF ZNZ          **MC REG SHOULD HAVE BEEN RESET
0F26  2800  BDIA 427                                   SET DR K=00        **ALLOW STOP ON ERROR
0F28  2020  BDIA 428                                   SET S6
0F2A  9324  BDIA 432                BCHK 011   START    BR
            BDIA 438    AEND
```

```
                                   ***********************************
                                   * CROSS REFERENCE FOR CSECT BDIA *
                                   ***********************************

BDIA 018     BDIA 008
BDIA 056     BDIA 053
BDIA 061     BDIA 061
BDIA 066     BDIA 066
BDIA 075     BDIA C72
BDIA 083     BDIA 080
BDIA 087     BDIA 084
BDIA 096     BDIA 096
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT BDIA *
                                  **********************************

BDIA 101     BDIA 101
BDIA 106     BDIA 103
BDIA 114     BDIA 111
BDIA 122     BDIA 054
BDIA 125     BDIA 119
BDIA 132     BDIA 132
BDIA 133     BDIA 133
BDIA 134     BDIA 134
BDIA 135     BDIA 135
BDIA 147     BDIA 147
BDIA 154     BDIA 154
BDIA 163     BDIA 163
BDIA 164     BDIA 164
BDIA 188     BDIA 188
BDIA 190     BDIA 190
BDIA 195     BDIA 137  BDIA 142  BDIA 183
BDIA 197     BDIA 202
BDIA 198     BDIA 196
BDIA 201     BDIA 166  BDIA 171  BDIA 176  BDIA 192
BDIA 207     BDIA 193
BDIA 211     BDIA 211
BDIA 212     BDIA 212
BDIA 216     BDIA 216
BDIA 217     BDIA 217
BDIA 228     BDIA 228
BDIA 229     BDIA 229
BDIA 231     BDIA 231
BDIA 239     BDIA 239
BDIA 240     BDIA 240
BDIA 242     BDIA 242
BDIA 245     BDIA 245
BDIA 248     BDIA 256
BDIA 255     BDIA 255
BDIA 259     BDIA 259
BDIA 261     BDIA 261
BDIA 269     BDIA 263  BDIA 269
BDIA 270     BDIA 270
BDIA 281     BDIA 277  BDIA 312  BDIA 324
BDIA 302     BDIA 283
BDIA 304     BDIA 304
BDIA 313     BDIA 302
BDIA 314     BDIA 314
BDIA 318     BDIA 315
BDIA 319     BDIA 319
BDIA 331     BDIA 317
BDIA 347     BDIA 347
BDIA 355     BDIA 343
BDIA 373     BDIA 354
BDIA 374     BDIA 376
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT BDIA *
                                  **********************************

BDIA 381     BDIA 375
BDIA 385     BDIA 394
BDIA 386     BDIA 382
BDIA 391     BDIA 391
BDIA 402     BDIA 402
BDIA 405     BDIA 405
BDIA 407     BDIA 407
BDIA 411     BDIA 411
BDIA 415     BDIA 400  BDIA 409
BDIA 418     BDIA 418
BDIA 423     BDIA 423
BDIA 426     BDIA 426
```

BMCK DESCRIPTIVE TEXT

WHEN A MACHINE ERROR OCCURS (SEE -MC- REGISTER), A MACHINE CHECK TRAP IS TAKEN IF -
MMSK 8 AND 9 NOT ON
MACHINE CHECK MASK LATCH ON (MW BIT 5)

IF MMSK 8 OR 9 ARE ON AND A MACHINE ERROR OCCURS, THE HARD STOP LATCH IS SET AND THE SYSTEM STOPS.

IF THE MACHINE CHECK TRAP IS ALLOWED THE TRAP ADDRESS 0220 IS FORCED ON THE SAR LINES AND THE -BMCK- ROUTINE IS ENTERED AT LABEL * TRAPWD *.
MMSK BITS 0-6 ARE TESTED AND IF ANY ARE ON (INDICATES THAT AN I/O TRAP OPERATION WAS IN PROGRESS WHEN THE MACHINE CHECK OCCURED), THE I/O OPERATION WILL BE TERMINATED, AND AN ATTEMPT TO CAUSE AN I/O INTERRUPT IS MADE. IF MMSK BITS 0-6 ARE NOT ON, A MACHINE CHECK INTERRUPT TAKES PLACE.

SINCE THE LOGOUT LATCH IS SET IN THIS ROUTINE, NO CPU INSTRUCTION WILL BE EXECUTED UNTIL THE PRINTOUT OF THE LOGOUT AREA IS COMPLETED. ANY PREVIOUS PR-KB OPERATION IN PROCESS, SUCH AS ALTER/DISPLAY, INSTRUCTION STEP ADDRESS TYPE- OUT, OR NORMAL PR-KB FUNCTIONS WILL BE FORCED TO TERMINATE. THE PRINT OUT OF THE LOGOUT AREA STARTS AFTER A CARRIAGE RETURN WITHOUT ANY HEADER INFORMATION.

THE -BCHK- ROUTINE LOADS THE LOGOUT AREA IN PROGRAM STORAGE, LOCATIONS 0080 - 0085, WITH THE INFORMATION LISTED BELOW. THE -ALDP- ROUTINE HANDLES THE PRINTOUT OF THIS LOGOUT AREA ON THE PR-KB.

MACHINE CHECK LOGOUT AREA

LOC 0080 — TRAP PRIORITY REGISTER (MMSK)

BIT
0- CHANNEL HIGH PRIORITY TRAP.
1- 2311 DISK CONTROL TRAP
2- CHANNEL LOW PRIORITY TRAP
3- 2540 READER TRAP
4- 2540 PUNCH TRAP
5- COMM. BIT SERVICE TRAP
6- COMM. CHAR. SERVICE TRAP
7- LEVEL 1 PRIORITY HOLD

LOC 0081 — BRANCH CONDITION REGISTER (BA)

BIT
0- CHANNEL 0 INTERRUPT LATCH
1- MODE BIT 0
2- MODE BIT 1
3- MODE BIT 2
4- IPL LATCH
5- LS ZONE BIT 0
6- LS ZONE BIT 1
7- LS ZONE BIT 2

LOC 0082 — MACHINE CHECK REGISTER (MC)

BIT
0- FILE CONTROL CHECK
1- STORAGE PROTECT PARITY CHECK
2- STORAGE ADDRESS PARITY CHECK
3- CONTROL WORD PARITY CHECK
4- STORAGE DATA PARITY CHECK
5- ALU ERROR CHECK
6- A-REG PARITY CHECK
7- B-REG PARITY CHECK

LOC 0083 — ERROR COUNT

LOC 0084-0085 — BACKUP ADDRESS-
ADDRESS OF THE CONTROL WORD IN OPERATION WHEN THE MACHINE CHECK OCCURRED.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                BMCK 001   T                          MACHINE CHECK TRAP ROUTINE
                BMCK 002   ATABLE  ADDR=0220
0220   3262     BMCK 003    TRAPWD                              LINK D MMSK9=1     SAVE ADDR+2 OF FAILING MICRO WD
0222   9028     BMCK 004                          006  GETOUT    BR                 GET OUT OF TRAP AREA
                BMCK 005   AEND
1028   7612     BMCK 006    GETOUT                              STH D DA,8A        TEMP STORE D REG FOR CHNL
102A   566F     BMCK 007                                        D0=MMSK            MOVE MMSK (TRAP PRIORITY) REG
102C   577F     BMCK 008                                        D1=BA              MOVE BA(BRANCH CONDITION) REG
102E   7672     BMCK 009                                        STH D DA,9E        SAVE MMSK & BA REGS IN AUX K-7
1030   2610     BMCK 010                                        SET BC K=01        SET LOGOUT LATCH
1032   A281     BMCK 011                          032  INTRAP N  BR IF MMSK=NZ      BR IF ANY MMSK BIT
1034   2400     BMCK 012    NOTRAP                              SET MODE K=00      SET TO CPU MODE & ZONE
1036   2135     BMCK 018                                        U1=0$K30           MACHINE CHECK OLD PSW ADDR
1038   B368     BMCK 019                    CSTS  106  PICKUP   BR                 GO TO STORE & LOAD MC PSW
1000   C693     BMCK 032    INTRAP 0        065  CHNLCK    BR IF MMSK0=1      BR IF IN CHNL HI PRIORITY TRAP
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
1002  D695  BMCK 036                 067  DSKCHN     BR IF MMSK1=1      BR IF IN DISK CHAINING TRAP
1004  E693  BMCK 040                 065  CHNLCK     BR IF MMSK2=1      BR IF IN CHNL LO PRIORITY TRAP
1006  F697  BMCK 044                 071  RDRTRP     BR IF MMSK3=1      BR IF IN READER TRAP
1008  C29B  BMCK 045                 073  PCHTRP     BR IF MMSK4=1      BR IF IN PUNCH TRAP
100A  021A  BMCK 058                            RST MMSK K=51          NO ICA, RST MMSK 5 (BIT) IF ON
100C  021C  BMCK 060                            RST MMSK K=61          NO ICA,RST MMSK 6(GENERAL) IF ON
100E  021E  BMCK 061                            RST MMSK K=71          RST LEVEL 1 PRIORITY
1010  9034  BMCK 063                 012  NOTRAP     BR                 LEVEL 1 PRIORITY, NOT IN TRAP
1012  98F8  BMCK 065  CHNLCK  DCLR 238  CCCXXX     BR                 CHANNEL TRAPS
1014  96AC  BMCK 067  DSKCHN  FILT 091  MCHCHK     BR
1016  2D02  BMCK 071  RDRTRP            SET RP K=10                   SET READER MACHINE CK LATCH
1018  B22E  BMCK 072           ETTR 043  RQUEST     BR                 GO BACK TO RDR TRAP ROUTINE
101A  2D40  BMCK 073  PCHTRP            SET RP K=04                   SET PUNCH MACHINE CK LATCH
101C  A4D8  BMCK 074           ETRP 071  INLK       BR                 GO BACK TO PCH TRAP ROUTINE
147A  2007  BMCK 092  LOGOUT            U0=0
147C  2400  BMCK 093                            SET MODE K=00          SET CPU MODE TO BR ON MC REG
147E  5E3F  BMCK 094                            V1=MC                  MOVE MC REG
1480  C4A7  BMCK 095                 120  NOTMC      BR IF Z=0          BR IF NOT MACHINE CK LOGOUT
1482  1212  BMCK 096                            RST MMSK K=91          RST MACHINE CHECK PRIORITY
            BMCK 097  *
            BMCK 098  *          FROM THIS POINT ON, A SECOND MACHINE CHECK WOULD CAUSE A RETURN TO THE
            BMCK 099  *          TRAP ADDRESS 0220. IF THE SECOND MACHINE CHECK OCCURRED PRIOR TO THE
            BMCK 100  *          LAST INSTRUCTION, THE CPU WILL STOP WITH THE CLOCK OFF. THE LAST
            BMCK 101  *          INSTRUCTION ALSO RESETS THE MC REGISTER.
            BMCK 102  *
1484  5672  BMCK 103                            RDH D DA,9E            RD SAVED MMSK & BA FROM AUX K-7
1486  2440  BMCK 104                            SET MODE K=04          BACK UP ZONE
1488  7672  BMCK 105                            STH D DA,9E            XFER MC BACKUP ADDR TO AUX K-7
148A  2400  BMCK 106                            SET MODE K=00          SET TO CPU MODE & ZONE
148C  2185  BMCK 107                            U1=0$K80               SET UP ADDR OF LOGOUT AREA
148E  420F  BMCK 108                            STP0=U0                DISABLE STORAGE PROTECT
1490  7618  BMCK 109                            STH D U+2              STORE MMSK, BA IN LOCATION 80,81
1492  7318  BMCK 110                            STB V1 U+1             STORE MACHINE CK REG IN 82
1494  5710  BMCK 111                            RDB D1 U+0             READ LAST ERROR COUNT FROM 83
1496  271B  BMCK 112                            D1=D1+K01              ADD 1 TO ERROR COUNT
1498  7718  BMCK 113                            STB D1 U+1             STORE UPDATED ERROR COUNT IN 83
149A  5672  BMCK 114                            RDH D DA,9E            READ BACKUP MICRO ADDR INTO D
149C  16C5  BMCK 115                            D0=D0*-KC0             STRIP BITS 0,1 - HZ,LZ
149E  1713  BMCK 116                            D1=D1*-K01             STRIP BIT 15 - ALU CARRY
14A0  6666  BMCK 117                            D=D-2                  MICRO ADDR POINTS AT FAILING WD
14A2  7618  BMCK 118                            STH D U+2              MICRO WD ADDR OF FAILURE IN 84,5
14A4  9248  BMCK 119           CCOM 182  RSTRKY     BAL                RESTORE THE CPU KEY
14A6  2404  BMCK 120  NOTMC                     SET MODE K=20          SET BACK TO 1052 MODE
14A8  2233  BMCK 121                            V0=0$K03               HW COUNT EQUALS 3
14AA  1F2E  BMCK 122                            RST TA K=F2  **RST RD, WT, M-FORCE, ALDP, SHARE IF ON
14AC  2171  BMCK 123                            U1=0-K07               U=00F8 - PRKB(1052) UCW ADDR
14AE  2807  BMCK 124                            T1=0
14B0  7A00  BMCK 125                            STH T AS,U+0           RST 1ST HW OF 1052 UCW
14B2  4632  BMCK 126                            RDH D DC,8E            READ OUT CONSTANT OF 4E08
14B4  261D  BMCK 127                            D0=D0+K10              SET D TO 5E18 FOR
14B6  271D  BMCK 128                            D1=D1+K10               'RDH H U+2'
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS

14B8     6662     BMCK 129                         STH D DC,9C          STORE 'DOIT' WD FOR ALDP ROUTINE
14BA     17AB     BMCK 130                         D1=D1<K0A            SET FLAGS FOR MC CK OR CE TRAP
14BC     2185     BMCK 131                         U1=0$K80             SET UP ADDR
14BE     88BC     BMCK 132           ALDP 241  LINEFD     BAL                  FOR A LINE FEED
14C0     9A8E     BMCK 133           ALDP 146  COLCNT     BR                   GO TO PRINT OUT LOG AREA
                                                ************************************
                                                * CROSS REFERENCE FOR CSECT BMCK *
                                                ************************************

BMCK 003     FILT 090
BMCK 006     BMCK 004
BMCK 012     BMCK 063
BMCK 032     BMCK 011
BMCK 065     BMCK 032   BMCK 040
BMCK 067     BMCK 036
BMCK 071     BMCK 044
BMCK 073     BMCK 045
BMCK 092     DYPE 021
BMCK 120     BMCK 095
```

BPSW DESCRIPTIVE TEXT

ENTRY POINTS TO THE BPSW ROUTINE

PROG — WHEN A PROGRAM INTERRUPTION IS DETECTED DURING INSTRUCTION PROCESSING THE BPSW ROUTINE IS ENTERED HERE. THE CURRENT PSW, ALONG WITH THE INTERRUPTION CODE, IS STORED IN THE PROGRAM OLD PSW LOCATION AND THE PROGRAM NEW PSW IS READ OUT AND PLACED IN CONTROL.

PROGA — ENTRY IS MADE HERE IF A PROTECTION ERROR OR AN EXECUTE INSTRUCTION IS DETECTED. PSW HANDLING IS THE SAME AS FOR PROG ENTRY.

ENTRYB — ENTRY IS MADE HERE FOR EXTERNAL INTERRUPT, CHANNEL INTERRUPT, OR CHANNEL INITIAL SELECTION. HIGH ORDER HEX DIGIT FOR PROPER PSW ADDRESS IS SET PRIOR TO ENTRY HERE.

ENTRY — A SUPERVISOR CALL OR MACHINE CHECK TRAP CAUSES ENTRY AT THIS LABEL.

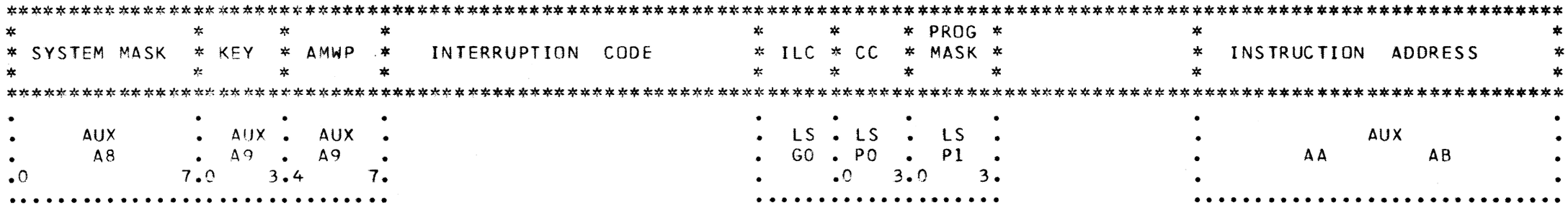
LPSW — ENTRY HERE IS TO COMPLETE THE 360 LOAD PSW INSTRUCTION OR TO LOAD THE IPL PSW.

INTBR — ENTRY HERE IS FROM CONDITIONAL BRANCH INSTRUCTIONS TO CHECK FOR ADDRESSING OR SPECIFICATION ERRORS.

INTXA
NINTXA — ENTRY TO EITHER OF THESE LABELS IS FROM THE BSWI ROUTINE WHEN THE SOFT STOP LOOP IS EXITED.

WAIT — THIS LABEL IS BRANCHED TO WHEN THE WAIT BIT IS TESTED AND FOUND TO BE ON.

PROGRAM STATUS WORD AND THE LOCATION OF THE CURRENT PSW



PSW ADDRESS ASSIGNMENTS

HEX		HEX	
00	INITIAL PROGRAM LOADING PSW		
18	EXTERNAL OLD PSW	58	EXTERNAL NEW PSW
20	SUPERVISOR CALL OLD PSW	60	SUPERVISOR CALL NEW PSW
28	PROGRAM OLD PSW	68	PROGRAM NEW PSW
30	MACHINE-CHECK OLD PSW	70	MACHINE-CHECK NEW PSW
38	INPUT/OUTPUT OLD PSW	78	INPUT/OUTPUT NEW PSW

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                BPSW 001    T                    PSW LOAD AND STORING                        001
                BPSW 002    *
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
09FC  2F07  BPSW 003      PROG                          H1=0               CLEAR INVALID ADD.
09FE  2125  BPSW 004      PROGA                         U1=0$K20           CREATE
0A00  21EB  BPSW 005      ENTRYB                        U1=U1+K0E           PROG. PSW
0A02  2007  BPSW 006      ENTRY                         U0=0                 ADDRESS
0A04  420F  BPSW 007      ENTRYA                        STP0=U0             STOR PROT DISABLE
0A06  5CE8  BPSW 008                                     H0=P0H             PLACE THE 4 BIT CC INTO H0 HIGH
0A08  3E45  BPSW 009                                     H0=H0$K40          ASSURE ILC = 1
0A0A  4DE5  BPSW 010                                     H0=P1XL$H0H        PLACE PROG MASK INTO H0 LOW
0A0C  E193  BPSW 011                          014  NOTEXC  BR IF S6=1         TEST FOR EXECUTE
0A0E  5892  BPSW 012                                     RDH I DA,AA        READ OUT EXECUTE INST. COUNTER
0A10  54B2  BPSW 013                                     RDH G DA,AE         RESTORE G REG
0A12  C416  BPSW 014      NOTEXC      016  WOW     BR IF G00=0        TEST OPCODE FORMAT FOR LENGTH
0A14  2E4D  BPSW 015                                     H0=H0+K40          ADD 1 TO LENGTH CODE  ILC = 2
0A16  D41A  BPSW 016      WOW         018  EXEC    BR IF G01=0        TEST OPCODE FORMAT FOR LENGTH
0A18  2E4D  BPSW 017      EXECA                         H0=H0+K40          ADD 1 TO LENGTH CODE ILC=2 OR 3
0A1A  781A  BPSW 018      EXEC                          STH I U-2          STORE INSTRUCTION ADDRESS
0A1C  7E1A  BPSW 019                                     STH H U-2          STORE ILC, CC, AND PROG. MASK
0A1E  7A1A  BPSW 020                                     STH T U-2          STORE INTERRUPT CODE
0A20  5E82  BPSW 021                                     RDH H DA,A8        READOUT SYS MASK, KEY, AMWP
0A22  7E10  BPSW 022                                     STH H U            STORE INTO OLD PSW
0A24  214D  BPSW 023                                     U1=U1+K40          ADJUST TO NEW PSW ADDRESS
0A26  5C18  BPSW 024      LPSW                          RDH P U+2          START NEW PSW READ
0A28  44CF  BPSW 025                                     SM=P0              SET SYSTEM MASK
0A2A  42DF  BPSW 027                                     STP0=P1            SET PROTECT KEY
0A2C  7C82  BPSW 029                                     STH P DA,A8        STORE SYS MASK, KEY, AND AMWP
0A2E  6004  BPSW 030                                     U=U+2                POINT TO 3RD HALFWORD OF PSW
0A30  5E18  BPSW 031                                     RDH H U+2            PLACE 3RD HALFWORD PSW IN H-REG
0A32  5810  BPSW 032                                     RDH I U              PLACE 4TH HALFWORD PSW IN I-REG
0A34  2020  BPSW 033                                     SET S6               SET EXECUTE IND. OFF
0A36  4ED3  BPSW 034                                     P1=H0XH+P1L          PLACE PROG MASK AND AMWP INTO P1
0A38  5ECB  BPSW 035                                     P0=H0H               PLACE ILC AND CC INTO P0 HIGH
0A3A  1CC5  BPSW 036                                     P0=P0*-KC0           ZERO OUT ILC
0A3C  FC40  BPSW 037                          039  NOT     BR IF P03=0          BRANCH IF CC IS 0 OR 2
0A3E  3C45  BPSW 038                                     P0=P0$K40            MAKE 4 BIT CC = TO 5 OR 7
0A40  2A07  BPSW 039      NOT                            T0=0
0A42  7CC2  BPSW 040                                     STH P DA,B8          SAVE P0-1
0A44  4FDF  BPSW 041                                     MW=P1                SET WAIT & MACHINE CK BITS
0A46  E955  BPSW 042                          066  WAIT    BR IF P16=1          BR IF WAIT BIT ON
0A48  2020  BPSW 054      INTBR                          SET S6               SET EXECUTE INDICATOR OFF
0A4A  F1D8  BPSW 055                          068  INTX    BR IF S7=0           BRANCH IF EXCEPTIONAL CONDITION
0A4C  AB71  BPSW 059      INTXA           069  ADDRER N  BR IF H1=NZ          BR IF 24 BITS ADDRESS
0A4E  09E3  BPSW 061                                     Z=I1*-K0E            TEST LOW ADDRESS BYTE
0A50  F0DA  BPSW 062                          064  SPEC    BR IF LZNZ           BRANCH IF ODD ADDRESS
0A52  AA06  BPSW 063                CICY 008  ISTART 3  BR                   BRANCH TO ICYCLES FOR INST. READ
0A5A  2B63  BPSW 064      SPEC                           T1=0$K06             SET BITS TO IDENTIFY SPEC. CHECK
0A5C  88E2  BPSW 065                          070  PROG1   BR                   BRANCH TO INCREMENT INST. ADDR.
0A54  0004  BPSW 066      WAIT                           RST S2           ** WAIT LOOP
0A56  F1D5  BPSW 067                          066  WAIT    BR IF S7=1       **     LEAVE WAIT LOOP ON INTERRUPT
0A58  8448  BPSW 068      INTX      BSWI 008  CHECKE  BR                       BRANCH TO DETERMINE INTERRUPT
08E0  2853  BPSW 069      ADDRER 0                       T1=0$K05             SET BITS TO IDENTIFY ADDR. ERROR
08E2  2407  BPSW 070      PROG1                          G0=0                 ZERO OP CODE REGISTER
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

08E4     6884     BPSW 071                                    I=I+2                  INCREMENT INSTRUCTION ADDRESS
08E6     89FE     BPSW 072                          004       PROGA      BR          BRANCH TO STORE INTERRUPT CODE
                                                              **********************************
                                                              * CROSS REFERENCE FOR CSECT BPSW *
                                                              **********************************
BPSW 003     CCOM 072  CDMD 061  CFAD 132  CFHA 037  CICY 057  CICY 071  CICY 078  CICY 080  CLST 015  CNVR 121  CSAS 130
BPSW 004     BPSW 072  BWRP 129  CBRC 064
BPSW 005     BSWI 131  DCHN 106
BPSW 006     CSTS 108
BPSW 014     BPSW 011
BPSW 016     BPSW 014
BPSW 018     BPSW 016
BPSW 024     BSYS 183  CSTS 007
BPSW 039     BPSW 037  BSYS 192
BPSW 054     CBRC 016
BPSW 059     BSWI 184
BPSW 064     BPSW 062
BPSW 066     BPSW 042  BPSW 067  BSWI 053  BSWI 188
BPSW 068     BPSW 055
BPSW 069     BPSW 059
BPSW 070     BPSW 065
```

BSTP DESCRIPTIVE TEXT

ENTRY TO THE -BSTP- ROUTINE IS MADE FROM THE -BSWI- ROUTINE WHEN AN INSTRUCTION STEP OPERATION OR A SET IC OPERATION IS PERFORMED. THE ENTRY IS MADE TO TEST THE 1052 AND DETERMINE IF THE INSTRUCTION COUNTER CAN BE TYPED OUT.

IF THE 1052 IS NOT BUSY, THE -ALDP- ROUTINE IS BRANCHED TO, AND THE INSTRUCTION COUNTER IS TYPED OUT.

IF THE 1052 IS BUSY, THE INSTRUCTION COUNTER IS NOT TYPED OUT. THE -BSWI- ROUTINE IS BRANCHED TO. THE CPU KEY, P-REG, AND I-REG ARE RESTORED AND THE -CICY- ROUTINE IS BRANCHED TO.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BSTP 001	T			INSTRUCTION TYPEOUT	
2ED6	2404	BSTP 002	TYPADD			SET MODE K=20	
2ED8	CFDF	BSTP 003		006	BUSY	BR IF TU0=1	CHECK AND BRANCH IF READ LATCH
2EDA	DFDF	BSTP 004		006	BUSY	BR IF TU1=1	CHECK AND BRANCH IF WRT LATCH
2EDC	CEE0	BSTP 005		007	OK	BR IF TT0=0	CHECK AND BRACH IF NO ATTEN
2EDE	AB20	BSTP 006	BUSY	BSWI 107	RESTRH	BR	NO TYPE OUT WILL OCCUR
2EE0	81AA	BSTP 007	OK	CCOM 174	STORH1	BAL	SAVE H1
2EE2	4E86	BSTP 008				H=I	PUT IC IN H REG
2EE4	26C3	BSTP 021				D0=0$K0C	BUILD
2EE6	2707	BSTP 023				D1=0	FLAGS
2EE8	2213	BSTP 024				V0=0$K01	SET UP COUNT
2EEA	9ACA	BSTP 025		ALDP 211	ENTRY	BR	GO DISPLAY

****
* CROSS REFERENCE FOR CSECT BSTP *
****

BSTP 002 BSWI 168
BSTP 006 BSTP 003 BSTP 004
BSTP 007 BSTP 005

BSWI DESCRIPTIVE TEXT

ENTRY POINTS

CHECKB - FROM I-CYCLES WHEN EXCEPTIONAL CONDITION, S7=0, IS RECOGNIZED. THE CONTROL WORD AT THIS ENTRY CREATES A VALID NEXT INSTRUCTION ADDRESS BY ENSURING THAT THE HIGH-ORDER BYTE IS CLEAR.

CHECKE - FROM LOAD PSW ROUTINE AFTER LOADING A PSW OR AFTER AN EFFECTIVE BRANCH INSTRUCTION. IF THE MODE SW IS IN THE INSN STEP POSITION, THE CONTROL WORD AT THIS ENTRY TURNS ON THE INSN STEP LATCH, BB BIT 5. BB5 IS TURNED OFF BY THE START KEY.

RESTRH - FROM ALTER/DISPLAY AND INTEGRATED I/O ROUTINES. THIS ENTRY RESTORES CPU MODE/ZONE, PROTECT KEY, AND L.S. REGISTERS BEFORE TESTING FOR THE NEXT INTERRUPT OR REQUEST. WHEN THE MICROPROGRAM EXITS FROM BSWI TO HANDLE A REQUEST, A RETURN IS MADE TO THIS ENTRY AFTER THE REQUEST IS PROCESSED.

DESCRIPTION

AFTER ENTRY AT EITHER -CHECKB- OR -CHECKE- , THE TIMER FUNCTION IS HANDLED. WHEN THE TIMER FEATURE IS PRESENT, THE TIMER IN PROGRAM STORAGE IS UPDATED IF THERE IS A VALUE IN THE -TIM- COUNTER. THE -TIM- COUNTER IS THE INVERSE OF THE HARDWARE TIMING COUNTER THAT IS STEPPED BY POWER SUPPLY PULSES. WHEN THE TIMER VALUE IN PROGRAM STORAGE IS REDUCED THROUGH ZERO, AN EXTERNAL INTERRUPT IS INITIATED BY SETTING BC BIT 1.

DESCRIPTION -- CONTINUED

THE FOLLOWING NOT S7 BRANCH CONDITIONS CAUSE ENTRY TO THE BSWI ROUTINE. THESE CONDITIONS ARE LISTED ACCORDING TO PRIORITY OF TESTING. EXCEPT FOR SOFT STOP, THE MICROPROGRAM BRANCHES TO THE APPROPRIATE ROUTINE FOR HANDLING THE CURRENT CONDITION.

1.	INTEGRATED I/O REQUEST	BB1 = 1
2.	EXTERNAL INTERRUPT MASKED	BB3 = 0
3.	CHANNEL 1 INTERRUPT MASKED	BB6 = 0
4.	COMMUNICATIONS CHANNEL INTERRUPT PENDING	BB7 = 0
5.	MASKED CHANNEL 0 INTERRUPT	BB2 = 0
6.	INSTRUCTION STEP LATCH	BB5 = 1
7.	SET IC LATCH	BB4 = 1
8.	SOFT STOP LATCH	BB0 = 0

THE SOFT STOP LATCH IS SET BY ANY ONE OF THE FOLLOWING -- INSN STEP LATCH, SET IC LATCH, ADDR MATCH, STOP SW, OR BY SETTING BC0. IF THE SOFT STOP LATCH IS ON WHEN TESTED, THE MICROPROGRAM LOOPS ON THE TESTS FOR INTEGRATED I/O REQUEST AND SET IC. EXIT FROM THE SOFT STOP LOOP OCCURS WHEN ONE OF THESE TESTS IS SUCCESSFUL OR WHEN THE SOFT STOP LATCH IS TURNED OFF. THE SOFT STOP LATCH IS TURNED OFF BY PRESSING THE START KEY, LOAD KEY, OR CSL KEY.

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS

             BSWI 001   T            EXCEPTIONAL CONDITION ROUTINE      D. L. SMITH
             BSWI 002   *            REGISTERS NOT TO BE DESTROYED  ARE
             BSWI 003   *             P0 - CONDITION CODE
             BSWI 004   *             P1 - PROG MASK  AMWP BITS
             BSWI 005   *             H1 - HI ORDER 8 BITS OF A 24 BIT BRANCH ADDRESS
             BSWI 006   *             I01- INSTRUCTION COUNTER
0446   2F07  BSWI 007    CHECKB                  H1=0                  CREATE VALID BRANCH ADDRESS
0448   3600  BSWI 008    CHECKE                  SET BC K=80         CONDITION INSTR STEP
044A   00C0  BSWI 009                            RST S K=0C           RST S4 AND S5
044C   81AA  BSWI 010                 CCOM 174  STORH1  BAL          STORE H1 IN AUX 00F4
044E   5AEF  BSWI 012                            H0=TIM                READ TIMER VALUE
0450   1EFF  BSWI 013                            H0=H0<KFF           INVERT VALUE
0452   C4F7  BSWI 014                      051   GOINA   BR IF Z=0          BR IF NO VALUE IN TIMER
0454   5E09  BSWI 024                            U0=H0                 CALCULATE
0456   6EE3  BSWI 025                            H0=H0+H0               VALUE
0458   6EE3  BSWI 026                            H0=H0+H0                FOR 60
045A   6E03  BSWI 027                            H0=H0+U0                 CYCLE
045C   2B55  BSWI 034                            T1=0$K50               READ OUT
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
045E  50B8    BSWI 035                                   RDH U T+2            CURRENT
0460  52BA    BSWI 036                                   RDH V T-2             VALUE
0462  2002    BSWI 037                                   SET S3           SET CARRY LATCH
0464  72E9    BSWI 038                                   V0C=V0-H0+C      DO
0466  71A9    BSWI 039                                   U1C=U1-T0+C           ARITH
0468  70A9    BSWI 040                                   U0C=U0-T0+C            FUNCTION ON TIMER
046A  42AF    BSWI 042                                   STP0=T0          DISABLE STORAGE PROTECT
046C  70B8    BSWI 044                                   STH U T+2            STORE
046E  72BA    BSWI 045                                   STH V T-2             RESULT
0470  9248    BSWI 047             CCOM 182  RSTRKY      BAL              RESTORE CPU KEY
0472  F5F9    BSWI 049                  052  CKWAIT     BR IF S3=1           BR IF NOT THRU 0
0474  2608    BSWI 050                                   SET BC K=40          TURN ON TIMER INT
0476  AB04    BSWI 051  GOINA           054  INA        BR               MAKE ADDRESSABLE
0478  E976    BSWI 052  CKWAIT         051  GOINA      BR IF P1 BIT6=0  BR IF NOT IN WAIT STATE
047A  8A54    BSWI 053             BPSW 066  WAIT       BR               TO AVOID RUNNING METER IN WAIT
2B04  DFE6    BSWI 054  INA            119  INB        BR IF BB1=0      BR IF NO NATIVE REQUEST
2B06  2040    BSWI 058                                   SET S5           SET FIRST TIME FLAG
2B08  2080    BSWI 059  INH                              SET S4           FOR REQUEST FROM BSWI - NOT SIO
2B0A  2406    BSWI 060                                   SET MODE K=30    SET READER MODE
2B0C  E990    BSWI 062                  068  TSTPCH     BR IF RS BIT6=0  BR IF NO READER REQUEST
2B0E  9F36    BSWI 064             ERRQ 030  RDST       BR               READER NATIVE REQUEST
2B10  EB94    BSWI 068  TSTPCH         082  TSTTWR     BR IF PS BIT6=0  BR IF NO PUNCH REQUEST
2B12  9F68    BSWI 070             ERRQ 017  PCHST      BR               PUNCH NATIVE REQUEST
2B14  2404    BSWI 082  TSTTWR                           SET MODE K=20    SET 1052 MODE
2B16  EA9A    BSWI 086                  100  TSTPTR     BR IF TT BIT6=0  BR IF NO 1052 REQUEST
2B18  8782    BSWI 088             DYPE 003  TWRREQ     BR               1052 REQUEST
2B1A  2482    BSWI 100  TSTPTR                           SET MODE K=18    SET 1403 PRINTER MODE
2B1C  FAA0    BSWI 102                  107  RESTRH     BR IF PRS BIT7=0 BR IF NO PRINTER REQUEST
2B1E  B0C4    BSWI 103             DPTQ 003  PTRREQ     BR               1403 PRINTER REQUEST
2B20  2400    BSWI 107  RESTRH                           SET MODE K=00    SET CPU MODE AND ZONE
2B22  2A07    BSWI 108                                   T0=0             RST T0 FOR NATIVES
2B24  9248    BSWI 110             CCOM 182  RSTRKY     BAL              RESTORE CPU KEY
2B26  E1AB    BSWI 112                  114  NOTPI      BR IF S6=1       BR IF P, I NOT STORED
2B28  8178    BSWI 113             CCOM 066  LRSTRB     BAL              RESTORE P, I
2B2A  C1E1    BSWI 114  NOTPI          116  NOTSIO     BR IF S4=1       BR IF FROM REQ AND NOT SIO
2B2C  8185    BSWI 115             CICY 005  ISTART N   N=S BITS67       RTN TO I CYCLES
2B60  8C7A    BSWI 116  NOTSIO       CCOM 187  RESTRH     BAL              RESTORE CPU BYTE H1
2B62  D1BC    BSWI 117                  153  IND        BR IF S5=0
2B64  0040    BSWI 118                                   RST S5
2B66  FFAF    BSWI 119  INB            132  INC        BR IF BB3=1          BR IF NO EXTERNAL INT.
2B68  59BF    BSWI 123                                   T1=XINT          READ INTERRUPT REG FOR INT CODE
2B6A  1BFF    BSWI 124                                   T1=T1<KFF        INVERT IT
2B6C  0640    BSWI 129                                   RST BC K=04
2B6E  2115    BSWI 130                                   U1=0$K10          INTERRUPT
2B70  8A00    BSWI 131             BPSW 005  ENTRYB     BR                FUNCTION
2B2E  EBB3    BSWI 132  INC            144  INE        BR IF BB6=1          BR IF NO CHAN 1  INTR.
2B30  A218    BSWI 134             DCLT 047  INTENT     BR               GO TO CHANNEL 1
2B32  FBB7    BSWI 144  INE            150  INF        BR IF BB7=1      BR IF NO COMMUN. CHNL INTERRUPT
2B34  5007    BSWI 148                                   STOP             COMMU INT WITH NO ICA INSTALLED
2B36  EFBB    BSWI 150  INF            152  ING        BR IF BB2=1          BR IF NO CHAN 0 INTR.
2B38  A3F2    BSWI 151             DCHN 041  ENTRY      BR               GO TO BURST CH 0 INT ROUTINE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

2B3A    DBC9    BSWI 152    ING          168   INJ        BR IF BB5=1            BR IF INSTR. STEP
2B3C    DF89    BSWI 153    IND          059   INH        BR IF BB1=1            BR IF NATIVE REQUEST
2B3E    CBCA    BSWI 162                 169   INI        BR IF BB4=0            BR IF NOT SET IC
2B40    2F07    BSWI 164                                  H1=0                    GET
2B42    508F    BSWI 165                                  IO=SWAB                  NEW
2B44    519F    BSWI 166                                  I1=SWCD                   INSTR.
2B46    0602    BSWI 167                                  RST BC K=10               ADDR
2B48    AED6    BSWI 168    INJ      BSTP 002  TYPADD     BR                     BR TO TYPE THE INST COUNTER
2B4A    CFBC    BSWI 169    INI          153   IND        BR IF BB0=0            CHECK FOR EXIT
2B4C    2404    BSWI 170                                  SET MODE K=20          PUT IN 1052 MODE
2B4E    FFDA    BSWI 171                 177   INK        BR IF TU3=0             ALT/DISP ACTIVE CHECK
2B50    3BE9    BSWI 172                                  T1=0-KE0               T=001F, ALDP BACKUP AREA
2B52    56AA    BSWI 173                                  RDH D AS,T-2           READ OUT ALDP FLAG HALF WORD
2B54    D258    BSWI 174                 176   STAROK     BR IF D0 BIT5=0        BR IF ALDP READY TO 'START'
2B56    88B4    BSWI 175             ALDP 271  RESET      BR                     GO TO SET SOFT STOP & MODE
2B58    1F2A    BSWI 176    STAROK                        RST TA K=D2 **RST RD, WR, ALDP ACTIVE LATCHES,SH.REQ.
2B5A    2400    BSWI 177    INK                           SET MODE K=00
2B5C    E903    BSWI 178                 188   INL        BR IF P16=1             WAIT STATE CHECK
2B5E    8A4C    BSWI 184             BPSW 059  INTXA      BR                     GO CHECK ADDRESS
2B02    8A54    BSWI 188    INL      BPSW 066  WAIT       BR                      GO TO WAIT
                                                ***********************************
                                                * CROSS REFERENCE FOR CSECT BSWI *
                                                ***********************************

BSWI 007     CICY 007
BSWI 008     BPSW 068
BSWI 051     BSWI 014  BSWI 052
BSWI 052     BSWI 049
BSWI 054     BSWI 051
BSWI 059     BSWI 153
BSWI 068     BSWI 062
BSWI 082     BSWI 068
BSWI 100     BSWI 086
BSWI 107     ALDP 261  ALDP 274  BSTP C06  BSWI 102  DPTQ 021  DPTR 077  DUMP 101  DYPE 052  DYPE 088  DYPE 125  ERRQ 062  EXFR 107
BSWI 114     BSWI 112
BSWI 116     BSWI 114
BSWI 119     BSWI 054
BSWI 132     BSWI 119
BSWI 144     BSWI 132
BSWI 150     BSWI 144
BSWI 152     BSWI 150
BSWI 153     BSWI 117  BSWI 169
BSWI 168     BSWI 152
BSWI 169     BSWI 162
BSWI 176     BSWI 174
BSWI 177     BSWI 171
BSWI 188     BSWI 178
```

BSYS DESCRIPTIVE TEXT

THE SYSTEM RESET ROUTINE IS ENTERED AT LABEL * SYSRST * FROM THE CHECK SUM ROUTINE -BCHK-, TO INITIALIZE CERTAIN I/O LATCHES AND AUXILIARY STORAGE LOCATIONS.

THIS ENTRY IS MADE AFTER EITHER THE SYSTEM RESET, LOAD, OR CONTROL STORAGE LOAD KEYS ARE PRESSED.

WHEN THE SYSTEM RESET KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

1. A HARDWARE RESET IS PERFORMED.
2. THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
3. THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
4. -BSYS- IS ENTERED AT LABEL * SYSRST *.

WHEN THE LOAD KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

1. A HARDWARE RESET IS PERFORMED.
2. THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
3. THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
4. -BSYS- IS ENTERED AT LABEL * SYSRST *.
5. THE -DCLL- ROUTINE IS BRANCHED TO FOR IPL START.
6. WHEN IPL IS COMPLETED, A BRANCH TO -BSYS- AT LABEL * RETURN *.
7. BRANCH TO -BPSW- TO LOAD THE INITIAL PSW.

WHEN THE CSL KEY IS PRESSED, THE FOLLOWING STEPS TAKE PLACE PRIOR TO ENTRY AT LABEL * SYSRST * -

1. A HARDWARE RESET IS PERFORMED.
2. THE -BCPL- ROUTINE IS EXECUTED.
3. THE RESIDENT DIAGNOSTIC -BDIA- IS EXECUTED.
4. THE CHECK SUM ROUTINE -BCHK- IS EXECUTED.
5. -BSYS- IS ENTERED AT LABEL * SYSRST *.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		BSYS 001	T			SYSTEM RESET	
228C	4E42	BSYS 002	SYSRST			RDH H DC,98	RD CONSTANT 00C8 INTO H REG
228E	2460	BSYS 004				SET MODE K=06	SET 2540 ZONE
2290	2607	BSYS 005				D0=0	RST 2540 FLAG REG
2292	24F6	BSYS 023				SET MODE K=3F	SET CHNL MODE AND ZONE
2294	4052	BSYS 024				RDH U DC,9A	RST U
2296	4206	BSYS 025				V=U	
2298	4406	BSYS 026				G=U	
229A	4606	BSYS 027				D=U	
229C	4A06	BSYS 028				T=U	
229E	2D08	BSYS 032				SET GB K=40	RST MULTIPLEX LATCH
22A0	2490	BSYS 042				SET MODE K=09	SET TO 2311 MODE AND ZONE
22A2	8F90	BSYS 043	GOBAL	FILE 288	SETDA	BAL	SET DEVICE ADDR & SELECT MODULE
22A4	A044	BSYS 044		FILE 344	RSTATT	BAL	GO RST GATED ATTENTION
22A6	49CF	BSYS 045				MS=P0	RESET MODULE SELECT
22A8	E5A3	BSYS 046			043 GOBAL	BR IF DASI2=1	BR IF ANY MORE GATED ATTENTION
22AA	2E08	BSYS 047				SET FIB K=40	RST FILE FLAG REG.
22AC	1E00	BSYS 048				RST FIB K=80	CHAIN END RESET
22AE	2482	BSYS 060				SET MODE K=18	SET 1403 MODE
22B0	2705	BSYS 061				D1=0$K00	SET CONSTANT ADDR TO 7000 & 900
22B2	2675	BSYS 063				D0=0$K70	SET PCCL CONSTANT ADDR FOR NO24K
22B4	5760	BSYS 067				RDB D1 AS,D+0	GET PCCL CONSTANT
22B6	497F	BSYS 068				PCCL=D1	SET PCCL LATCH FOR 1403
22B8	2F00	BSYS 069				SET PRB K=00	RESET TO DIAGNOSTIC DECODE 1
22BA	2400	BSYS 070				SET MODE K=00	SET CPU MODE AND ZONE
22BC	6EE4	BSYS 071				H=H+2	H=00CA
22BE	7AEA	BSYS 072				STH T AS,H-2	CLEAR 1403 SENSE AND STATUS
22C0	7AE0	BSYS 074	LOOP1			STH T AS,H+0	ZERO OUT 1ST HALFWORD OF THE

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
22C2  2F1D    BSYS 075                                   H1=H1+K10            NATIVE DEVICES' UCW
22C4  F4C0    BSYS 076                 074   LOOP1       BR IF AC=0             (IN 00C8, 00D8, 00E8, 00F8)
22C6  2F91    BSYS 077                                   H1=0-K09           H=00F6
22C8  7AEA    BSYS 078                                   STH T AS,H-2       CLEAR 1052 SENSE BYTE
22CA  2E15    BSYS 093                                   H0=0$K10           H=10F4
22CC  7AEA    BSYS 094                                   STH T AS,H-2       RST PCH INDICATORS & SENSE BYTE
22CE  7AEA    BSYS 095                                   STH T AS,H-2       RST PCH OLD STACKER SELECT
22D0  7AE0    BSYS 096                                   STH T AS,H+0       RST RDR INDICATORS & SENSE BYTE
22D2  7A32    BSYS 115                                   STH T DA,8E  **RST STATUS/ACT BYTE FOR CHNL 1 OR 2311
22D4  7A52    BSYS 117                                   STH T DA,9A        RST 2311 SENSE HALF WORD 1
22D6  7A62    BSYS 118                                   STH T DA,9C        RST 2311 SENSE HALF WORD 2
22D8  40A6    BSYS 173                                   U=T
22DA  3212    BSYS 174                                   SET MMSK K=91      SET MCH CK TRAP BIT, STOP IF ERR
22DC  1210    BSYS 175                                   RST MMSK K=81      RST SYS RST BIT TO ALLOW TRAPS
22DE  20A0    BSYS 176                                   SET S K=0A
22E0  C3E4    BSYS 177                 179   SYSTEM      BR IF BA4=0        BR IF NOT IPL
22E2  B390    BSYS 178               DCLL 002  IPL        BR
22E4  E8EE    BSYS 179   SYSTEM         184   RESPSW      BR IF DR6=0          BR IF NOT PSW RESTART
22E6  062C    BSYS 180   RETURN                           RST BC K=62
22E8  1212    BSYS 181                                   RST MMSK K=91
22EA  4052    BSYS 182                                   RDH U DC,9A        ZERO OUT U REG
22EC  8A26    BSYS 183               BPSW 024  LPSW       BR
22EE  162C    BSYS 184   RESPSW                          RST BC K=E2        SET STOP,RST CSL,RST SYS,RST IPL
22F0  1212    BSYS 185                                   RST MMSK K=91                RST TRAP
22F2  5C82    BSYS 186                                   RDH P DA,A8          LOAD
22F4  44CF    BSYS 187                                   SM=P0                  UP
22F6  42DF    BSYS 188                                   STP0=P1                THE
22F8  4852    BSYS 189                                   RDH I DC,9A            LAST
22FA  5CC2    BSYS 190                                   RDH P DA,B8              PSW
22FC  2F07    BSYS 191                                   H1=0               ZERO HIGH ORDER ADDR
22FE  8A40    BSYS 192                     BPSW 039  NOT  BR
              BSYS 193   ATABLE  ADDR=00B4
00B4  0E0C    BSYS 196   C                              XAUX'0E0CCD1F'
              BSYS 206   AEND
              BSYS 207   ATABLE  ADDR=00F4
00F4  0090    BSYS 209   C                              XAUX'0090'
              BSYS 215   AEND
              BSYS 216   ATABLE  ADDR=0004
0004  0090    BSYS 218   C                              XAUX'0090'
              BSYS 222   AEND
```

****
* CROSS REFERENCE FOR CSECT BSYS *
****

```
BSYS 002    BCHK 104
BSYS 043    BSYS 046
BSYS 074    BSYS 076
BSYS 179    BSYS 177
BSYS 180    DCLL 045
BSYS 184    BSYS 179
```

BWRP DESCRIPTIVE TEXT

ENTRY POINT

THIS ROUTINE IS ENTERED WHEN A TRAP ADDRESS OF 0210 IS FORCED BY --
1. STORAGE PROTECT VIOLATION. PROTECT KEYS DO NOT MATCH.
2. STORAGE WRAP VIOLATION. THIS OCCURS WHEN A PROGRAM STORAGE ADDRESS THAT EXCEEDS THE SYSTEM SIZE IS ENCOUNTERED.

SYSTEM SIZE	PROG STOR ADDR RANGE
16K	0000 - 3FFF
24K	0000 - 5FFF
32K	0000 - 7FFF
48K	0000 - BFFF

DESCRIPTION

THE UNCONDITIONAL BRANCH WORD AT THE START OF THE ROUTINE STORES THE ADDRESS +2 OF THE TRAPPED WORD IN BITS 2-6 OF THE I-REG BACKUP, I.E., ZONE 4. BIT 7 OF THE I-REG BACKUP IS SET FROM DYN BIT 3. IF THE TRAP OCCURS DURING INSTRUCTION READOUT IN I-CYCLES, THE ADDRESS STORED IN THE BACKUP I-REG IS THE ADDRESS OF THE TRAPPED WORD +3 BECAUSE DYN BIT 3 IS SET TO 1 LATER IN THE ROUTINE, THIS BACKUP ADDRESS IS DECREMENTED BY 1 TO POINT TO THE CONTROL WORD THAT FOLLOWS THE TRAPPED WORD. THIS CONTROL WORD IS USED TO DETERMINE WHICH HALFWORD OF THE INSTRUCTION WAS BEING READ WHEN THE TRAP OCCURRED. THIS IS DONE IN ORDER TO SET UP THE NECESSARY REGISTERS FOR THE PENDING INTERRUPT.

AFTER THE UNCONDITIONAL BRANCH, DYN BIT 1 IS TESTED. THIS BIT IS SET TO 1 FOR A STORAGE WRAP VIOLATION. IF DYN1 = 0, A STORAGE PROTECT VIOLATION HAS OCCURRED. FOR EITHER VIOLATION, THE ROUTINE DETERMINES THE MODE BY TESTING BA BITS 1,2, AND 3. FOR FILE, COMMUNICATIONS, OR CHANNEL MODES, THE MICROPROGRAM BRANCHES TO THE APPROPRIATE ROUTINE TO ATTEMPT THE INTERRUPT. FOR THE INTEGRATED I/O UNITS REMAINING, EITHER PROTECTION CHECK OR PROGRAM CHECK IS SET IN THE CHANNEL STATUS OF THE APPROPRIATE UCW, THEN THE INTERRUPT IS ATTEMPTED.

WHEN A STORAGE PROTECT VIOLATION IN CPU MODE OCCURS, THE PROTECTION ERROR IS FLAGGED, AND THE PROGRAM INTERRUPT IS INITIATED.

WHEN A STORAGE WRAP VIOLATION IN CPU MODE OCCURS, TESTS ARE MADE TO DETERMINE IF THE WRAP OCCURRED DURING INSTRUCTION READOUT, AS PREVIOUSLY STATED. FOR ANY STORAGE WRAP VIOLATION IN CPU MODE, THE ROUTINE EXITS TO FLAG THE ADDRESS ERROR AND INITIATE THE INTERRUPT.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              BWRP 001    T              WRAP TRAP AND STORAGE                D. L. SMITH
              BWRP 002    *
              BWRP 003    *
              BWRP 004    *
              BWRP 005    *
              BWRP 006    *
              BWRP 007    ATABLE  ADDR=0210
0210    887C  BWRP 008                    011  NXTADR     BR                   SAVE BACKUP ADDR IF IN CPU MODE
              BWRP 009    AEND
087C    D4FA  BWRP 011     NXTADR         050  PROTCK     BR IF DYN1=0         BR IF STORAGE PROTECT
087E    979D  BWRP 012                    016  STRG   N   N=BA BITS123         FIND THE MODE CAUSING WRAP TRAP
0E80    F790  BWRP 016     STRG   0       083  ADDRCK     BR IF BA3=0
0E82    96B4  BWRP 018     STRG   1 FILT 095  WRAP       BR                   2311 MODE
0E84    8F64  BWRP 024     STRG   2 ERRQ 086  RPHUCW     BAL                  SET D TO 00E8
0E86    904A  BWRP 030     STRG   3       140  WRAP       BR                   1403 MODE
0E88    904A  BWRP 032     STRG   4       140  WRAP       BR                   1052 MODE
0E8A    5007  BWRP 040     STRG   5                  STOP       ** WRAP IN ICA MODE WITH NO ICA INSTALLED
0E8C    904A  BWRP 042     STRG   6       140  WRAP       BR                   2540 READER WRAP
0E8E    B490  BWRP 046     STRG   7 DCLR 134  WRAP       BR                   CHANNEL
087A    97F5  BWRP 050     PROTCK         051  PRT    N   N=BA BITS123         FIND THE MODE CAUSING THE PROT.
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0AE0  9D74  BWRP 051     PRT   0       124   CPUPRT     BR                   CPU MODE
0AE2  96B0  BWRP 053     PRT   1  FILT 093   PRTCHK     BR                  2311 MODE
0AE4  B1FC  BWRP 057     PRT   2       133   PROTEC     BR                  2540 PUNCH PROTECT CK
0AE6  B1FC  BWRP 061     PRT   3       133   PROTEC     BR                  1403 MODE
0AE8  B1FC  BWRP 065     PRT   4       133   PROTEC     BR                  1052 MODE
0AEA  5007  BWRP 073     PRT   5                          STOP    ** PROT CK IN ICA MODE WITH NO ICA INSTALLED
0AEC  B1FC  BWRP 075     PRT   6       133   PROTEC     BR                  2540 READER PROTECT CK
0AEE  98D6  BWRP 079     PRT   7  DCLR 136   PROTCH     BR                  CHANNEL
0E90  2A07  BWRP 083     ADDRCK             TO=0
0E92  E1BC  BWRP 091                 121   NXTD       BR IF S6=0
0E94  2440  BWRP 093                                   SET MODE K=04        PUT IN BACKUP ZONE
0E96  78F2  BWRP 094                                   STH I DA,BE           SAVE ADDRESS STORED
0E98  2400  BWRP 095                                   SET MODE K=00        PUT BACK IN CPU ZONE
0E9A  56F2  BWRP 096                                   RDH D DA,BE           GET ADDRESS IN TO WORK ON
0E9C  16C5  BWRP 097                                   D0=D0*-KC0
0E9E  371B  BWRP 098                                   D1=D1-K01
0EA0  4660  BWRP 099                                   RDH D CS,D            READ CONTROL WORD FOLLOWING
0EA2  06BD  BWRP 100                                   Z=D0<KB0             CHECK
0EA4  E0AC  BWRP 101                 106   NJXB       BR IF HZNZ            FOR
0EA6  C64B  BWRP 102                                   Z=D0<K04               FIRST
0EA8  F0AC  BWRP 103                 106   NJXB       BR IF LZNZ              INSTRUCTION
0EAA  F339  BWRP 104                 119   NXTC       BR IF D17=1              READOUT
0EAC  08CD  BWRP 106     NJXB                          Z=I0<KC0             CHECK ADDRESS
0EAE  C4BA  BWRP 114                 120   NXTB       BR IF ZNZ            DURING INSTRUCTION
0EB0  092B  BWRP 115                                   Z=I1<K02              READOUT
0EB2  C4BA  BWRP 116                 120   NXTB       BR IF ZNZ
0EB4  E23B  BWRP 117                 120   NXTB       BR IF D06=1          BR IF THIRD READOUT FAILED
0EB6  6886  BWRP 118                                   I=I-2                BUILD IT AS AN RR OP FLAGGED
0EB8  4452  BWRP 119     NXTC                          RDH G DC,9A          RESET ILC CODE AND G1 REG
0EBA  AA20  BWRP 120     NXTB       CICY 056   ADDERR C  BR
0EBC  8176  BWRP 121     NXTD       CCOM 065   LRSTRA     BAL                 RESTORE LOCAL STORAGE REGS
0EBE  AA20  BWRP 122                CICY 056   ADDERR 0  BR
1D74  E1F9  BWRP 124     CPUPRT           126   NOSAVE     BR IF  S6=1         RESTORE REGISTERS
1D76  8174  BWRP 125                CCOM 064   LRSTR      BAL                       IF S6 IS ON
1D78  2A07  BWRP 126     NOSAVE                          TO=0                 FLAG
1D7A  2B43  BWRP 127                                     T1=0$K04              PROTECTION
1D7C  2F07  BWRP 128                                     H1=0                   ERROR
1D7E  89FE  BWRP 129                BPSW 004   PROGA      BR
            BWRP 130     *                                ********************************************
            BWRP 131     *                                NATIVE I/O WRAP AND PROTECT
            BWRP 132     *                                ********************************************
31FC  2B95  BWRP 133      PROTEC                          T1=0$K90             SET PROTECT CHK & SECONDARY
31FE  904C  BWRP 134                 142   WRORPR     BR
104A  2BA5  BWRP 140      WRAP                           T1=0$KA0             SET PROGRAM CHK & SECONDARY
104C  2607  BWRP 142      WRORPR                         D0=0                 FIX D0 FOR 2540
104E  8F78  BWRP 146                ERRQ 081   RDSTCH     BAL                 READ CHNL STATUS
1050  64B5  BWRP 147                                     G0=G0$T1             OR IN PROGRAM CHK OR PROTECT CHK
1052  2543  BWRP 148                                     G1=0$K04             RESET CHAINING--KEEP ACTIVE ON
1054  2EC3  BWRP 149                                     H0=0$KCC              SET CH END DEV END UNIT STATUS
1056  8761  BWRP 150                 152   WHICHN N   N=D1 BITS23          BREAK OUT NATIVES
1060  8D68  BWRP 152     WHICHN 0   DPTR 075   TAKEIN     BR                  1403              * TRY
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

1062     B01E     BWRP 156   WHICHN 1  DCLR 052  NTYPE      BR                   2540 READER      *  TO
1064     B01E     BWRP 157   WHICHN 2  DCLR 052  NTYPE      BR                   2540 PUNCH        *  CAUSE INT
1066     8792     BWRP 162   WHICHN 3  DYPE 011  NOATEN     BR                    1052 CAUSE INTERUPT
                                                 ***********************************
                                                 * CROSS REFERENCE FOR CSECT BWRP *
                                                 ***********************************

BWRP 011     BWRP 008
BWRP 016     BWRP 012
BWRP 050     BWRP 011
BWRP 051     BWRP 050
BWRP 083     BWRP 016
BWRP 106     BWRP 101  BWRP 103
BWRP 119     BWRP 104
BWRP 120     BWRP 114  BWRP 116  BWRP 117
BWRP 121     BWRP 091
BWRP 124     BWRP 051
BWRP 126     BWRP 124
BWRP 133     BWRP 057  BWRP 061  BWRP 065  BWRP 075
BWRP 140     BWRP 030  BWRP 032  BWRP 042
BWRP 142     BWRP 134
BWRP 152     BWRP 150
```

CBIN DESCRIPTION

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP19	COMPARE	19	R1 R2		
RROP1A	ADD	1A	R1 R2		
RROP1B	SUBTRACT	1B	R1 R2		
RROP1E	ADD LOGICAL	1E	R1 R2		
RROP1F	SUBTRACT LOGICAL	1F	R1 R2		
RXOP49	COMPARE HALFWORD	49	R1 X2	B2 D2	D2
RXOP4A	ADD HALFWORD	4A	R1 X2	B2 D2	D2
RXOP4B	SUBTRACT HALFWORD	4B	R1 X2	B2 D2	D2
RXOP59	COMPARE	59	R1 X2	B2 D2	D2
RXOP5A	ADD	5A	R1 X2	B2 D2	D2
RXOP5B	SUBTRACT	5B	R1 X2	B2 D2	D2
RXOP5E	ADD LOGICAL	5E	R1 X2	B2 D2	D2
RXOP5F	SUBTRACT LOGICAL	5F	R1 X2	B2 D2	D2

OBJECTIVES

ADD - THE SECOND OPERAND IS ADDED TO THE FIRST OPERAND, AND THE SUM IS PLACED IN THE FIRST OPERAND LOCATION.

ADD LOGICAL - SAME AS ADD. THE CONDITION CODES DIFFER.

SUBTRACT - THE SECOND OPERAND IS SUBTRACTED FROM THE FIRST OPERAND, AND THE DIFFERANCE IS PLACED IN THE FIRST OPERAND LOCATION.

SUBT. LOG. - SAME AS SUBT. THE CONDITION CODES DIFFER

COMPARE - THE FIRST OPERAND IS COMPARED WITH THE SECOND OPERAND, AND THE RESULT DETERMINES THE CONDITION CODE. THE COMPARE IS DONE BY COMPLEMENT ADDING THE OPERANDS.

GENERAL OBJECTIVES

1. FETCH OPERANDS
2. ADD OR COMPLEMENT ADD OPERANDS
3. SET CONDITION CODE
4. RETURN TO I-CYCLES

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
                 CBIN 001     T                      BINARY ADD AND SUB COMP
                 CBIN 002     *
                 CBIN 003     *
                 CBIN 004     *                                                    ** RR COMPARE
14C2     8270    CBIN 005      RROP19    CCOM 036  CRGET      BAL                    FETCH OPERANDS
14C4     AD54    CBIN 006                     009  CXCOMP     BR                     GO COMPLEMENT ADD OPERANDS
                 CBIN 007     *                                                    ** RX COMPARE
2D52     B3F6    CBIN 008      RXOP59    CCOM 045  CXGET      BAL                    FETCH OPERANDS
2D54     B460    CBIN 009      CXCOMP    CCOM 086  CSLOOP     BAL                    COMPLEMENT ADD OPERANDS
2D56     E4DA    CBIN 010                     012  OVER       BR IF NOVFL            BR IF NO OVERFLOW
2D58     168D    CBIN 011                          D0=D0<K80                         INVERT SIGN
2D5A     F5CE    CBIN 012      OVER           061  CONDC0     BR IF S2=0             BR IF RESULT ZERO -- EQUAL
2D5C     C63B    CBIN 013                     064  CONDC1     BR IF D00=1            BR IF RESULT IS MINUS -- A-LOW
2D5E     2C25    CBIN 014                          P0=0$K20                          SET CONDITION CODE 2 -- A-HIGH
2D60     8185    CBIN 015                CICY 005  ISTART N   N=S BITS67            RETURN TO I-CYCLES
                 CBIN 016     *
                 CBIN 017     *                                                    ** RX COMPARE HALFWORD
23EE     AE06    CBIN 018      RXOP49    CCOM 012  CXGETH     BAL                    FETCH OPERANDS, EXTEND SIGN
23F0     AD54    CBIN 019                     009  CXCOMP     BR                     GO COMPLEMENT ADD OPERANDS
                 CBIN 020     *
                 CBIN 021     *                                                    ** RR ADD
24FC     8270    CBIN 022      RROP1A    CCOM 036  CRGET      BAL                    FETCH OPERANDS
24FE     AC1A    CBIN 023                     027  CXADD      BR                     GO ADD OPERANDS
                 CBIN 024     *
                 CBIN 025     *                                                    ** RX ADD
2C18     B3F6    CBIN 026      RXOP5A    CCOM 045  CXGET      BAL                    FETCH OPERANDS
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
2C1A  B462    CBIN 027    CXADD    CCOM 087  CXLOOP     BAL                  ADD OPERANDS
2C1C  E4BD    CBIN 028    CAC1BC   CLST 009  OVCHEK     BR IF OVFL           BR IF OVERFLOW TO SET CC3
2C1E  E5B0    CBIN 029             CLST 034  NEGCC0     BR IF S2=0           BR IF ZERO RESULT TO SET CC0
2C20  C625    CBIN 030    CCCHK    CLST 036  NEGCC1     BR IF D00=1          BR IF NEG SIGN TO SET CC1
2C22  AC46    CBIN 031             CLST 024  TESCC2     BR                   BR TO SET CONDITION CODE 2
              CBIN 032    *
              CBIN 033    *                                          ** RX ADD HALFWORD
08E8  AE06    CBIN 034    RXOP4A   CCOM 012  CXGETH     BAL                  FETCH OPERANDS
08EA  AC1A    CBIN 035             CCOM 027  CXADD      BR                   GO ADD OPERANDS
              CBIN 036    *
              CBIN 037    *                                          ** RR SUBTRACT
277C  8270    CBIN 038    RROP1B   CCOM 036  CRGET      BAL                  FETCH OPERANDS
277E  88DC    CBIN 039             CCOM 043  CXSUB      BR                   GO COMPLEMENT ADD OPERANDS
              CBIN 040    *
              CBIN 041    *                                          ** RX SUBTRACT
08DA  B3F6    CBIN 042    RXOP5B   CCOM 045  CXGET      BAL                  FETCH OPERANDS
08DC  B460    CBIN 043    CXSUB    CCOM 086  CSLOOP     BAL                  COMPLEMENT ADD OPERANDS
08DE  AC1C    CBIN 044             CCOM 028  CAC1BC     BR                   GO TEST FOR CONDITION CODE
              CBIN 045    *
              CBIN 046    *                                          ** RX SUBTRACT HALFWORD
169C  AE06    CBIN 047    RXOP4B   CCOM 012  CXGETH     BAL                  FETCH OPERANDS
169E  88DC    CBIN 048             CCOM 043  CXSUB      BR                   GO COMPLEMENT ADD OPERANDS
              CBIN 049    *
              CBIN 050    *                                          ** RR ADD LOGICAL
2E78  8270    CBIN 051    RROP1E   CCOM 036  CRGET      BAL                  FETCH OPERANDS
2E7A  AD44    CBIN 052             CCOM 056  CXADDL     BR                   GO ADD OPERANDS
              CBIN 053    *
              CBIN 054    *                                          ** RX ADD LOGICAL
2D42  B3F6    CBIN 055    RXOP5E   CCOM 045  CXGET      BAL                  FETCH OPERANDS
2D44  B462    CBIN 056    CXADDL   CCOM 087  CXLOOP     BAL                  GO ADD OPERANDS
2D46  76A8    CBIN 057    S2S3CK                        STH D AS,T+2         PUT RESULT IN
2D48  7EAA    CBIN 058                                  STH H AS,T-2           FIRST OPERAND LOCATION
2D4A  E5B9    CBIN 059             CCOM 063  NONZER     BR IF S2=1           BR IF RESULT NOT ZERO
2D4C  F5B5    CBIN 060             CCOM 068  ZRCARY     BR IF S3=1           BR IF 0-BIT CARRY
2D4E  2C07    CBIN 061    CONDC0                        P0=0                 SET CONDITION CODE 0
2D50  8185    CBIN 062             CICY 005  ISTART N   N=S BITS67           RETURN TO I-CYCLES
2D38  F5BF    CBIN 063    NONZER   CCOM 066  NZCARY     BR IF S3=1           BR IF ZERO-BIT CARRY
2D3A  2C55    CBIN 064    CONDC1                        P0=0$K50             SET CONDITION CODE 1
2D3C  8185    CBIN 065             CICY 005  ISTART N   N=S BITS67           RETURN TO I-CYCLES
2D3E  2C75    CBIN 066    NZCARY                        P0=0$K70             SET CONDITION CODE 3
2D40  8185    CBIN 067             CICY 005  ISTART N   N=S BITS67           RETURN TO I-CYCLES
2D34  2C25    CBIN 068    ZRCARY                        P0=0$K20             SET CONDITION CODE 2
2D36  8185    CBIN 069             CICY 005  ISTART N   N=S BITS67           RETURN TO I-CYCLES
              CBIN 070    *
              CBIN 071    *                                          ** RR SUBTRACT LOGICAL
2078  8270    CBIN 072    RROP1F   CCOM 036  CRGET      BAL                  FETCH OPERANDS
207A  8E7C    CBIN 073             CCOM 077  CXSUBL     BR                   GO COMPLEMENT ADD OPERANDS
              CBIN 074    *
              CBIN 075    *                                          ** RX SUBTRACT LOGICAL
0E7A  B3F6    CBIN 076    RXOP5F   CCOM 045  CXGET      BAL                  FETCH OPERANDS
0E7C  B460    CBIN 077    CXSUBL   CCOM 086  CSLOOP     BAL                  COMPLEMENT ADD OPERANDS
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

0E7E     AD46    CBIN 078                057  S2S3CK    BR                    GO STORE RESULT & SET COND CODE
                                              ************************************
                                              * CROSS REFERENCE FOR CSECT CBIN *
                                              ************************************

CBIN 005     CICY 178
CBIN 008     CICY 210
CBIN 009     CBIN 006  CBIN 019
CBIN 012     CBIN 010
CBIN 018     CICY 194
CBIN 022     CICY 179
CBIN 026     CICY 211
CBIN 027     CBIN 023  CBIN 035
CBIN 028     CBIN 044
CBIN 034     CICY 195
CBIN 038     CICY 180
CBIN 042     CICY 212
CBIN 043     CBIN 039  CBIN 048
CBIN 047     CICY 196
CBIN 051     CICY 183
CBIN 055     CICY 215
CBIN 056     CBIN 052
CBIN 057     CBIN 078
CBIN 061     CBIN 012
CBIN 063     CBIN 059
CBIN 064     CBIN 013
CBIN 066     CBIN 063
CBIN 068     CBIN 060
CBIN 072     CICY 184
CBIN 076     CICY 216
CBIN 077     CBIN 073
```

BRANCH INSTRUCTIONS

CBRC DESCRIPTION

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP05	BRANCH AND LINK	05	R1 R2		
RROP06	BRANCH ON COUNT	06	R1 R2		
RROP07	BR ON CONDITION	07	M1 R2		
RXOP44	EXECUTE	44	R1 X2	B2 D2	D2
RXOP45	BRANCH AND LINK	45	R1 X2	B2 D2	D2
RXOP46	BRANCH ON COUNT	46	R1 X2	B2 D2	D2
RXOP47	BR ON CONDITION	47	M1 X2	B2 D2	D2
RSOP86	BR ON INDEX HIGH	86	R1 R3	B2 D2	D2
RSOP86	BRANCH ON INDEX LOW OR EQUAL	87	R1 R3	B2 D2	D2

OBJECTIVES

BRANCH AND LINK - STORE THE UPDATED RIGHTMOST 32 BITS OF THE CURRENT PSW, AND SET THE BRANCH ADDRESS INTO THE INSTRUCTION COUNTER.

BRANCH ON COUNT - BRANCH TO DEVELOPED ADDRESS, EACH TIME, UNTIL SUM IN R1 EQUALS ZERO. EACH TIME THROUGH, SUBTRACT 1 FROM VALUE IN R1.

OBJECTIVES CONTINUED

BR ON CONDITION - IF CONDITION CODE AND MASK MATCH, BRANCH TO DEVELOPED ADDRESS.
NOTE - THE CONDITION CODE IN LS REG P0 IS CODED AS FOLLOWS.

COND CODE	P0 HIGH
0	0
1	5
2	2
3	7

EXECUTE - BITS 8-15 OF THE SUBJECT INSTRUCTION ARE MODIFIED BY OR-ING THEM WITH BYTE 3 OF R1. THEN THE MODIFIED INSTRUCTION IS EXECUTED. THE SUBJECT INSTRUCTION IS LOCATED AT THE 2ND OPERAND EFFECTIVE ADDRESS.

BR ON INDEX HIGH- ADD THE CONTENTS OF R3 TO R1. COMPARE THIS SUM TO THE CONTENTS OF R3, OR R3 PLUS 1 IF R3 IS EVEN. IF COMPARE IS HIGH, BRANCH TO THE EFFECTIVE SECOND OPERAND ADDRESS.

BRANCH ON INDEX
LOW OR EQUAL - SAME AS BRANCH ON INDEX HIGH, EXCEPT THE BRANCH IS ON LOW OR EQUAL COMPARE.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

                CBRC 001   T                    BRANCH INSTRUCTIONS                D. L. SMITH
                CBRC 002   *
                CBRC 003   *
                CBRC 004   *                  BRANCH CONDITION RR
293A   55B3     CBRC 005    RROP07                          T1=G1XH            EXECUTION
293C   C483     CBRC 006                        013  LEG   1 BR IF Z=0          OF
293E   5EA8     CBRC 007                                    RDH H AS,T+2         RR
2940   50A0     CBRC 008                                    RDH U AS,T+0          BCR
                CBRC 009   *
                CBRC 010   *                  BRANCH CONDITION RX
2942   9C03     CBRC 011    RXOP47            012  LEG   N N=P0 BITS123      EXECUTION
2900   C507     CBRC 012    LEG    0          015  LEG   3 BR IF G10=1        OF
2902   8185     CBRC 013    LEG    1   CICY 005  ISTART N N=S BITS67           RX AND
2904   E502     CBRC 014    LEG    2          013  LEG   1 BR IF G12=0          RR
2906   4806     CBRC 015    LEG    3                      I=U                    BRANCH
2908   8A48     CBRC 016    LEG    4   BPSW 054  INTBR     BR                     CONDITION
290A   D507     CBRC 017    LEG    5          015  LEG   3 BR IF G11=1
290C   8185     CBRC 018    LEG    6   CICY 005  ISTART N N=S BITS67
290E   F507     CBRC 019    LEG    7          015  LEG   3 BR IF G13=1
2910   8185     CBRC 020                   CICY 005  ISTART N N=S BITS67
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                CBRC 021   *
                CBRC 022   *            BRANCH AND LINK  RR
2912   0040     CBRC 023    RROP05                        RST S5               EXECUTION
2914   55B3     CBRC 024                                  T1=G1XH            RR
2916   C49F     CBRC 025                  032  OP45A      BR IF Z=0           BALR
2918   5EA8     CBRC 026                                  RDH H AS,T+2
291A   50A0     CBRC 027                                  RDH U AS,T
                CBRC 028   *
                CBRC 029   *
                CBRC 030   *            BRANCH AND LINK RX
291C   2040     CBRC 031    RXOP45                        SET S5               EXECUTION
291E   5C2B     CBRC 032    OP45A                         V0=P0H             RR
2920   3245     CBRC 033                                  V0=V0$K40            AND
2922   E1B6     CBRC 034                  044  EXECUT     BR IF S6=0           RX
2924   D428     CBRC 035                  037  OP45B      BR IF G01=0            BRANCH
2926   224D     CBRC 036    OP45C                         V0=V0+K40              AND
2928   4D25     CBRC 037    OP45B                         V0=P1XL$V0H            LINK
292A   2307     CBRC 038                                  V1=0
292C   55BB     CBRC 039                                  T1=G1H
292E   72A8     CBRC 040                                  STH V AS,T+2
2930   78A0     CBRC 041                                  STH I AS,T
2932   D187     CBRC 042                  015  LEG    3   BR IF S5=1
2934   8185     CBRC 043              CICY 005  ISTART N   N=S BITS67
2936   5892     CBRC 044    EXECUT                        RDH I DA,AA        RESTORE EXECUTE INST. COUNTER
2938   A926     CBRC 045                  036  OP45C      BR
                CBRC 046   *               EXECUTE OP CODE          44
2D88   AB25     CBRC 047    RXOP44    CICY 056  ADDERR N   BR IF H1=NZ
2D8A   01E3     CBRC 048                                  Z=U1*-K0E          SPEC CHECK
2D8C   F086     CBRC 049                  065  SPEC       BR IF LZNZ
2D8E   E182     CBRC 050                  063  OP44B      BR IF S6=0         EXECUTE CHECK
2D90   7CC2     CBRC 051                                  STH P DA,B8        SAVE P
2D92   7892     CBRC 052                                  STH I DA,AA
2D94   74B2     CBRC 053                                  STH G DA,AE        SAVE G REG
2D96   4806     CBRC 054                                  I=U
2D98   0020     CBRC 055                                  RST S6
2D9A   55BB     CBRC 056                                  T1=G1H
2D9C   C481     CBRC 057                  066  RETRNC     BR IF Z=0
2D9E   5498     CBRC 058                                  RDH G I+2          READ INSTRUCTION
2DA0   2B3B     CBRC 059                                  T1=T1+K03          ADJUST REG ADDRESS
2DA2   53A0     CBRC 060                                  RDB V1 AS,T+0
2DA4   6535     CBRC 061                                  G1=G1$V1
2DA6   B463     CBRC 062              CICY 013  OPHI   N   N=G0H
2D82   2B33     CBRC 063    OP44B                         T1=0$K03
2D84   89FE     CBRC 064              BPSW 004  PROGA      BR
2D86   AE94     CBRC 065    SPEC      CICY 070  SPECHK     BR
2D80   AA06     CBRC 066    RETRNC    CICY 008  ISTART 3   BR
                CBRC 067   *            BRANCH ON COUNT OP CODE   06
2DA8   0040     CBRC 068    RROP06                        RST S5
2DAA   55B3     CBRC 069                                  T1=G1XH
2DAC   C4B5     CBRC 070                  075  OP4X       BR IF Z=0
2DAE   5EA8     CBRC 071                                  RDH H AS,T+2
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

2DB0  50A0    CBRC 072                                   RDH U AS,T+0
              CBRC 073    *          BRANCH ON COUNT OP CODE   46
2DB2  2040    CBRC 074     RXOP46                        SET S5
2DB4  55BB    CBRC 075     OP4X                          T1=G1H
2DB6  56A8    CBRC 076                                   RDH D AS,T+2
2DB8  52AA    CBRC 077                                   RDH V AS,T-2
2DBA  C00A    CBRC 078                                   RST S K=50
2DBC  73A9    CBRC 079                                   V1C=V1-T0+C          DECREMENT
2DBE  72A9    CBRC 080                                   V0C=V0-T0+C           COUNT
2DC0  77A9    CBRC 081                                   D1C=D1-T0+C            BY
2DC2  76A9    CBRC 082                                   D0C=D0-T0+C             ONE
2DC4  76A8    CBRC 083                                   STH D AS,T+2
2DC6  72AA    CBRC 084                                   STH V AS,T-2
2DC8  D5CD    CBRC 085                 087  CBRANC       BR IF S1=1           CHECK RESULT
2DCA  8185    CBRC 086     NBRANC   CICY 005  ISTART N   N=S BITS67
2DCC  D1CA    CBRC 087     CBRANC        086  NBRANC      BR IF S5=0           COUNT ONLY GO TO ICYC
2DCE  A906    CBRC 088                 015  LEG    3    BR
              CBRC 089    *          BRANCH INDEX HI,LO,EQUAL    THE FIRST WORD OF THIS ROUTINE IS ENTERED
              CBRC 090    *                                      BY OP CODES 86 AND 87
2C88  9816    CBRC 091     RSOP86   CCOM 058  LSAVEB     BAL                  SAVE I AND P
2C8A  55BB    CBRC 092                                   T1=G1H
2C8C  58A8    CBRC 093                                   RDH I AS,T+2
2C8E  52AA    CBRC 094                                   RDH V AS,T-2         INDEX IN IV
2C90  55B3    CBRC 095                                   T1=G1XH              GET
2C92  56A8    CBRC 096                                   RDH D AS,T+2          INCREMENT
2C94  5CAA    CBRC 097                                   RDH P AS,T-2           FROM R3
2C96  6D3B    CBRC 098                                   P1C=P1+V1            ADD
2C98  6C2D    CBRC 099                                   P0C=P0+V0+C           INDEX
2C9A  679D    CBRC 100                                   D1C=D1+I1+C            INTO
2C9C  668D    CBRC 101                                   D0C=D0+I0+C             INCREMENT
2C9E  3B15    CBRC 102                                   T1=T1$K10
2CA0  58A8    CBRC 103                                   RDH I AS,T+2         GET
2CA2  52AA    CBRC 104                                   RDH V AS,T-2          COMPARAND
2CA4  55BB    CBRC 105                                   T1=G1H               SET R1 ADDRESS
2CA6  76A8    CBRC 106                                   STH D AS,T+2          STORE
2CA8  7CAA    CBRC 107                                   STH P AS,T-2           SUM
2CAA  C004    CBRC 108                                   RST S2
2CAC  3002    CBRC 109                                   SET S K=90
2CAE  7D3D    CBRC 110                                   P1C=P1%V1+C          COMPARE
2CB0  7C2D    CBRC 111                                   P0C=P0%V0+C           COMPARAND
2CB2  779D    CBRC 112                                   D1C=D1%I1+C            TO THE
2CB4  768D    CBRC 113                                   D0C=D0%I0+C             SUM
2CB6  8178    CBRC 114                 CCOM 066  LRSTRB  BAL                  RESTORE I,P
2CB8  E481    CBRC 115                      119  OVERFL    BR IF OVFL           BR IF OVERFLOW
2CBA  F583    CBRC 116     CHKANS           120  SIGN      BR IF S2=1
2CBC  F007    CBRC 117     LHCHEK           122  GOBR      BR IF GO BIT7=1      EQUAL FIRST LO 2ND
2CBE  8185    CBRC 118     NOBR     CICY 005  ISTART N   N=S BITS67
2C80  168D    CBRC 119     OVERFL                         D0=D0<K80            OVERFLOW INVERT SIGN
2C82  C63D    CBRC 120     SIGN             117  LHCHEK    BR IF D0 BIT0=1
2C84  F03F    CBRC 121                      118  NOBR      BR IF GO BIT7=1      SUM IS HIGH
2C86  A906    CBRC 122     GOBR             015  LEG    3  BR
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS
                                                  ***********************************
                                                  * CROSS REFERENCE FOR CSECT CBRC *
                                                  ***********************************

CBRC 005     CICY 140
CBRC 011     CICY 192
CBRC 012     CBRC 011
CBRC 013     CBRC 006  CBRC 014
CBRC 015     CBRC 012  CBRC 017  CBRC 019  CBRC 042  CBRC 088  CBRC 122
CBRC 023     CICY 138
CBRC 031     CICY 190
CBRC 032     CBRC 025
CBRC 036     CBRC 045
CBRC 037     CBRC 035
CBRC 044     CBRC 034
CBRC 047     CICY 189
CBRC 063     CBRC 050
CBRC 065     CBRC 049
CBRC 066     CBRC 057
CBRC 068     CICY 139
CBRC 074     CICY 191
CBRC 075     CBRC 070
CBRC 086     CBRC 087
CBRC 087     CBRC 085
CBRC 091     CICY 223  CICY 224
CBRC 117     CBRC 120
CBRC 118     CBRC 121
CBRC 119     CBRC 115
CBRC 120     CBRC 116
CBRC 122     CBRC 117
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT              COMMENTS
                CCOM 001   T                                 **** COMMON ROUTINES  *****
                CCOM 002   *
                CCOM 003   *
                CCOM 004   *
                CCOM 005   *
                CCOM 006   *            **                    **THE FOLLING ROUTINES**
                CCOM 007   *            **                   **ARE OPERAND FETCHES**
                CCOM 008   *
                CCOM 009   *
                CCOM 010   *
                CCOM 011   *                                  ** LINKED GET HALF ROUTINE **
2E06  AB25      CCOM 012    CXGETH    CICY 056   ADDERR N    BR IF H1=NZ            CHECK FOR VALID ADDRESS
2E08  C1E3      CCOM 013                                      Z=U1*-KCE
2E0A  F084      CCOM 014                  027    CXSPEC      BR IF LZNZ             CHECK BOUNDARY
2E0C  5E10      CCOM 015                                      RDH H U+0
2E0E  CE01      CCOM 016                  025    CXNEG       BR IF H0 BIT0=1        CHECK SIGN
2E10  2305      CCOM 017                                      V1=0$K00               EXTEND PLUS SIGN
2E12  5329      CCOM 018    CCSIGN                            V0=V1
2E14  40E6      CCOM 019                                      U=H
2E16  55BB      CCOM 020    GETR1                             T1=G1H
2E18  56A8      CCOM 021                                      RDH D AS,T+2
2E1A  5EAA      CCOM 022                                      RDH H AS,T-2
2E1C  100E      CCOM 023                                      RST  S K=F0                RESET S BITS 0,1,2,3
2E1E  128E      CCOM 024                                      RTN
2E00  23E7      CCOM 025    CXNEG                             V1=0$KFF
2E02  C492      CCOM 026                  018    CCSIGN      BR IF ZNZ              UNC BR ON ZNZ
2E04  AE94      CCOM 027    CXSPEC    CICY 070   SPECHK      BR
                CCOM 028   *                                 ** OPERAND FETCH FOR LOADS **
0AC8  55B3      CCOM 029    LGET                              T1=G1XH
0ACA  56A8      CCOM 030                                      RDH D AS,T+2
0ACC  5EAA      CCOM 031                                      RDH H AS,T-2
0ACE  55BB      CCOM 032                                      T1=G1H                 SET DESTINATION ADDR
0AD0  0008      CCOM 033                                      RST S1                     RESET ZNZ  STAT
0AD2  128E      CCOM 034                                      RTN
                CCOM 035   *                                 ** RR OPERAND FETCH **
0270  55B3      CCOM 036    CRGET                             T1=G1XH
0272  52A8      CCOM 037                                      RDH V AS,T+2           2ND OPERAND
0274  50AA      CCOM 038                                      RDH U AS,T-2
0276  55BB      CCOM 039                                      T1=G1H
0278  56A8      CCOM 040                                      RDH D AS,T+2           1ST OPERAND
027A  5EAA      CCOM 041                                      RDH H AS,T-2
027C  100E      CCOM 042                                      RST  S K=F0                RESET S BITS 0,1,2,3
027E  128E      CCOM 043                                      RTN
                CCOM 044   *                                 ** RX OPERAND FETCH **
33F6  5218      CCOM 045    CXGET                             RDH V U+2              2ND OPERAND
33F8  5010      CCOM 046                                      RDH U U+0
33FA  55BB      CCOM 047                                      T1=G1H
33FC  56A8      CCOM 048                                      RDH D AS,T+2           1ST OPERAND
33FE  5EAA      CCOM 049                                      RDH H AS,T-2
3400  100E      CCOM 050                                      RST  S K=F0                RESET S BITS 0,1,2,3
3402  128E      CCOM 051                                      RTN
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

            CCOM 052    *
            CCOM 053    *
            CCOM 054    *
            CCOM 055    *                                 ** LOCAL STORE SAVE **
1812  70A2  CCOM 056     LSAVE                 STH U DA,AC          SAVE U
1814  74B2  CCOM 057     LSAVEA                STH G DA,AE               G
1816  7CC2  CCOM 058     LSAVEB                STH P DA,B8          SAVE P
1818  E19E  CCOM 059                    062    LSAVEC     BR IF S6=0           TEST EXECUTE-BR IF YES
181A  7892  CCOM 060                               STH I DA,AA               I
181C  0020  CCOM 061                               RST S6
181E  128E  CCOM 062     LSAVEC                    RTN
            CCOM 063    *                                 ** RESTORE LOCAL STORE **
0174  50A2  CCOM 064     LRSTR                 RDH U DA,AC          RESTORE  U
0176  54B2  CCOM 065     LRSTRA                RDH G DA,AE                   G
0178  5CC2  CCOM 066     LRSTRB                RDH P DA,B8                   P
017A  5892  CCOM 067     LRSTRC                RDH I DA,AA                   I
017C  2020  CCOM 068                               SET S6
017E  128E  CCOM 069                               RTN
            CCOM 070    *
1BDC  2B73  CCOM 071     FLGDEX                    T1=0$K07             FLAG DATA
1BDE  89FC  CCOM 072                    BPSW 003   PROG       BR         EXECPTION
            CCOM 073    *                                 ** LINKED COMPLIMENT ROUTINE **
            CCOM 074    *
            CCOM 075    *
33DA  2002  CCOM 076     COMPL                     SET S3               SET CARRY
33DC  7FFB  CCOM 077                               H1C=0-H1+C
33DE  7EEB  CCOM 078                               H0C=0-H0+C
33E0  777B  CCOM 079                               D1C=0-D1+C
33E2  766B  CCOM 080                               D0C=0-D0+C
33E4  100A  CCOM 081                               RST  S K=D0               RESET ZNZ,CARRY,COMP
33E6  128E  CCOM 082                               RTN
            CCOM 083    *                                    ** COMMON ADD,SUBTRACT,COMPARE ROUTINE **
            CCOM 084    *
            CCOM 085    *
3460  3002  CCOM 086     CSLOOP                    SET  S K=90               SET COMPLIMENT,CARRY
3462  7F1D  CCOM 087     CXLOOP                    H1C=H1%U1+C
3464  7E0D  CCOM 088                               H0C=H0%U0+C
3466  773D  CCOM 089                               D1C=D1%V1+C
3468  762D  CCOM 090                               D0C=D0%V0+C
346A  128E  CCOM 091                               RTN
            CCOM 092    *                  ********************************************************8********
            CCOM 093    *
            CCOM 094    *
            CCOM 095    *                           ***  CSW STORE ROUTINE ********
            CCOM 096    *
            CCOM 097    *
200E  2607  CCOM 110     FULCSW                    D0=0
2010  577B  CCOM 114                               D1=D1H                ADJUST UCW ADDR TO NAT KEY LOC
2012  3763  CCOM 115                               D1=D1$K06             D=UCW ADDR-00X6  ANY NATIVE
2014  5B60  CCOM 116                               RDB T1 AS,D+0         GET NATIVE KEY T1=KKKK0000
2016  5B6B  CCOM 117                               D0=T1H                D0=KKKK0000-NATIVE KEY-NOT COMM
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
2018  2707    CCOM 118     HELP                         D1=0               ZERO OUT ADDRESS BYTE FOR CSW
201A  F0A5    CCOM 120                127   CSWADR     BR IF LZ=0         UNC BR
201C  4E52    CCOM 121     CLEAR                        RDH H DC,9A        ENTRY FOR ALL ZERO CSW STORE
              CCOM 122    *                                                  READ ZERO CONSTANT INTO H
201E  4252    CCOM 123     STATOS                       RDH V DC,9A  ** ENTRY FOR STAT STORE-ZERO REST OF CSW
              CCOM 124    *                                                  READ ZERO CONSTANT INTO V
2020  4626    CCOM 125                                  D=V
2022  4026    CCOM 126                                  U=V
2024  2B45    CCOM 127     CSWADR                       T1=0$K40
2026  42AF    CCOM 129                                  STPO=T0            DIXABLE CPU KEY
2028  76B8    CCOM 131                                  STH D T+2           KEY AND 0,S
202A  72B8    CCOM 132                                  STH V T+2              NEXT CCW ADDR
202C  6AA4    CCOM 133                                  T=T+2
202E  70B0    CCOM 134                                  STH U T+0          RES COUNT
2030  2B47    CCOM 135     STATUS                       T1=0$K44           ENTRY FOR STATUS ONLY
2032  42AF    CCOM 137                                  STPO=T0            DIXABLE CPU KEY
2034  7EB0    CCOM 139                                  STH H T+0          STATUS=UNIT,CHAN.
2036  5282    CCOM 141                                  RDH V DA,A8        PUT CPU KEY BACK IN Q. IF IN
2038  423F    CCOM 142                                  STPO=V1             FILE ZONE TO & T1 REGS ARE
              CCOM 143    *                                                    ACCESSED INSTEAD OF V0 & V1.
203A  C5BE    CCOM 145                147   NORTN      BR IF S0=0         S0=1 FOR RTN FROM THIS ROUTINE
203C  128E    CCOM 146                                  RTN
203E  AED0    CCOM 147     NORTN       158   CC1B       BR                 GO RST MODE ,SET CC1
              CCOM 148    *
              CCOM 149    *
              CCOM 150    *
              CCOM 151    *
              CCOM 152    *
              CCOM 153    *                                  ****COMMON CONDITION CODE SET****
              CCOM 154    *
010C  2400    CCOM 155     CC0B                             SET MODE K=C0
010E  AA74    CCOM 156                CICY 003  CHECKX     BR
              CCOM 157    *
2ED0  2400    CCOM 158     CC1B                             SET MODE K=00
2ED2  2C55    CCOM 159     CC1C                             PO=0$K50                SET CC1-BITS2,3=COND CODE
2ED4  8185    CCOM 160                CICY 005  ISTART N   N=S BITS67
              CCOM 161    *
2ECA  2400    CCOM 162     CC2B                             SET MODE K=00
2ECC  2C25    CCOM 163     CC2C                             PO=0$K20                SET CC2
2ECE  8185    CCOM 164                CICY 005  ISTART N   N=S BITS67
              CCOM 165    *
2DF6  2400    CCOM 166     CC3B                             SET MODE K=00
2DF8  2C75    CCOM 167     CC3C                             PO=0$K70                SET CC3
2DFA  8185    CCOM 168                CICY 005  ISTART N   N=S BITS67
              CCOM 169    *
              CCOM 170    *          ************************************************************
              CCOM 171    *
              CCOM 172    *                            **  STORE H1 ROUTINE  **
              CCOM 173    *
01AA  2BB1    CCOM 174     STORH1                           T1=0-K0B
01AC  7FA0    CCOM 175                                      STB H1 AS,T+0
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

01AE    128E     CCOM 176                                    RTN
                 CCOM 177   *
                 CCOM 178   *          ************************************************************
                 CCOM 179   *
                 CCOM 180   *                                  **  RESTORE CPU KEY ROUTINE  **
                 CCOM 181   *
1248    5682     CCOM 182    RSTRKY                           RDH D DA,A8            GET CPU KEY
124A    427F     CCOM 183                                     STP0=D1                RESTORE CPU KEY INTO Q
124C    128E     CCOM 184                                     RTN
                 CCOM 185   *                                  ****************************************
                 CCOM 186   *                                  ***  RESTORE H1 ROUTINE ***
0C7A    2BB1     CCOM 187    RESTRH                           T1=0-K0B               T=00F4
0C7C    5FA0     CCOM 188                                     RDB H1 AS,T+0
0C7E    128E     CCOM 189                                     RTN
                                                  ***********************************
                                                  * CROSS REFERENCE FOR CSECT CCOM *
                                                  ***********************************

CCOM 012     CBIN 018  CBIN 034  CBIN 047  CMLT 014
CCOM 018     CCOM 026
CCOM 020     CNVR 013
CCOM 025     CCOM 016
CCOM 027     CCOM 014
CCOM 029     CDVD 004  CLST 005  CLST 018  CLST 030  CLST 040
CCOM 036     CBIN 005  CBIN 022  CBIN 038  CBIN 051  CBIN 072  CLOG 047  CLOG 078  CLOG 114  CLOG 144  CMLT 011
CCOM 045     CBIN 008  CBIN 026  CBIN 042  CBIN 055  CBIN 076  CLOG 050  CLOG 082  CLOG 118  CLOG 148  CMLT 004
CCOM 056     CDMD 009  DUMP 003  ERDR 067
CCOM 057     CDVD 011  CFCY 016  CFCY 036  CMLT 015  CNVR 009
CCOM 058     CBRC 091  DPTS 048  DYPE 005  DYPE 114  DYPE 237  ERRQ 022  ERRQ 066
CCOM 062     CCOM 059
CCOM 064     BWRP 125  CDMD 054  CDMD 059  CDMD 139  DUMP 100
CCOM 065     BWRP 121  CDVD 086  CDVD 098  CMLT 074  CNVR 111
CCOM 066     BSWI 113  CBRC 114  CFAD 072  CNVR 047  DPTR 072  DPTS 149  ERDR 070
CCOM 067     CFAD 127  CFCY 068  CFHA 033  CFLS 042
CCOM 071     CDMD 055  CNVR 073  CSAS 025  CTRT 085
CCOM 076     CDVD 018  CLST 007  CLST 032  CLST 042  CNVR 017
CCOM 086     CBIN 009  CBIN 043  CBIN 077  CLOG 051
CCOM 087     CBIN 027  CBIN 056
CCOM 110     DCHN 102  DPTT 041
CCOM 123     DCHN 120  DCLC 057  DPTT 015  DPTT 054  FILE 129  FINT 039
CCOM 127     CCOM 120  DCLT 040  FILE 018  FINT 009
CCOM 135     DCLA 286  DCLC 185  DCLR 011  DPTS 150  DPTT 068  FILE 140  FILX 467
CCOM 147     CCOM 145
CCOM 155     DCLA 041  DCLA 287  DCLC 065  DCLH 021  DPTT 011  DYPE 213  ERRQ 012  FILE 030  FILE 098  FILX 219  FILX 355
CCOM 158     CCOM 147  DCLA 042
CCOM 162     DCLA 066  DCLA 369  DCLB 005  DCLT 008  DPTT 025  FILE 026
CCOM 166     DCLC 037  DYPE 226
CCOM 174     BSTP 007  BSWI 010
CCOM 182     BMCK 119  BSWI 047  BSWI 110  CSTS 058
CCOM 187     BSWI 116  DCLT 041
```

CDMD -- DECIMAL MULTIPLY-DIVIDE ROUTINES

ENTRY POINT

SSOPFC - THIS IS THE COMMON ENTRY POINT FOR BOTH MULTIPLY DECIMAL AND DIVIDE DECIMAL.

MULT DECIMAL	FC	L1	L2	B1 D1	D1 D1	B2 D2	D2 D2
DIV DECIMAL	FD	L1	L2	B1 D1	D1 D1	B2 D2	D2 D2

OBJECTIVES

MULTIPLY - THE PRODUCT OF THE MULTIPLIER AND THE MULTIPLICAND REPLACES THE MULTIPLICAND IN THE 1ST OPERAND LOCATION.
THE MULTIPLIER SIZE IS LIMITED TO 15 DIGITS AND SIGN -- LENGTH CODE OF 7 IN L2. ALSO, L2 MUST NOT BE LARGER THAN OR EQUAL TO L1.
THE MULTIPLICAND MUST HAVE HIGH-ORDER ZERO BYTES EQUAL TO OR GREATER THAN THE NUMBER OF BYTES IN THE MULTIPLIER FIELD.
THE MAXIMUM PRODUCT SIZE IS 31 DIGITS. AT LEAST ONE HIGH-ORDER DIGIT OF THE PRODUCT FIELD IS ZERO.

DIVIDE - THE DIVIDEND IS DIVIDED BY THE DIVISOR. THE QUOTIENT AND THE REMAINDER REPLACE THE DIVIDEND IN THE 1ST OPERAND LOCATION.
THE QUOTIENT FIELD IS PLACED IN THE LEFT PORTION OF THE DIVIDEND FIELD. THE REMAINDER IS THE SAME SIZE AS THE DIVISOR AND OCCUPIES THE LOW-ORDER BYTES OF THE DIVIDEND FIELD.
THE DIVISOR SIZE IS LIMITED TO 15 DIGITS AND SIGN -- LENGTH CODE OF 7 IN L2. ALSO, L2 MUST NOT BE LARGER THAN OR EQUAL TO L1.

DESCRIPTION

THE ROUTINE CHECKS OPERAND SIZE AND SETS THE SIGN VALUE. THEN, TWO COPIES OF THE MULTIPLIER OR DIVISOR ARE STORED.

	AUX STORAGE FIELD	POINTER FOR LOW-ORDER BYTE
STRAIGHT COPY	0018-001F	00BC
SKEWED COPY	0038-00BF	00BE

DESCRIPTION CONTINUED

MULTIPLY OP

1. TEST FOR ENOUGH ZEROES IN MULTIPLICAND FIELD.
2. SHIFT THE MOST SIGNIFICANT BYTE AND THE FOLLOWING BYTES OF THE MULTIPLICAND TO THE HIGH-ORDER POSITIONS OF THE MULTIPLICAND FIELD. SUPPLY ZEROES FOR THE VACATED BYTES, REMOVE SIGN.
3. STARTING WITH THE LOW-ORDER, TEST EACH DIGIT OF MULTIPLICAND. TRUE ADD IF UNDER 5, SUBTRACT IF OVER 4. ENTER CALCULATE LOOP FOR EACH DIGIT. FOR THE 1ST DIGIT, THE STRAIGHT MULTIPLIER WITH 90 AS THE HIGH-ORDER BYTE IS T/C ADDED TO THE PRODUCT FIELD. THE 9 IS IN ALIGNMENT WITH THE TESTED MULTIPLICAND DIGIT. TIC ADD CYCLES ARE CONTINUED UNTIL TEST DIGIT IS 0 FOR ADD OR 9 FOR SUBTRACT.
ALTERNATE MULTIPLICAND DIGITS USE THE SKEWED MULTIPLIER WITH 09 AS THE HIGH-ORDER BYTE UNLESS THE TEST DIGIT GOES TO 0 FOR ADD OR 9 FOR SUBTRACT. FOR THIS CASE, THE MULTIPLIER IS NOT SWITCHED.
4. DO SPECIAL SKEWED ADD IF NECESSARY, INSERT SIGN, AND END THE OPERATION.

DIVIDE OP

1. DO A TRIAL SUBTRACT TO DETERMINE IF THE QUOTIENT AND REMAINDER CAN BE CONTAINED IN THE DIVIDEND FIELD. SKEWED DIVISOR IS USED. A CARRY OUT OF THE HIGH-ORDER POSITION INDICATES A DIVIDE CHECK. IF DIVIDE CHECK OCCURS, SKEWED DIVISOR IS ADDED BACK TO RESTORE THE DIVIDEND AND ROUTINE EXITS TO TAKE PROGRAM INTERRUPT.
2. FOR 1ST CALCULATE CYCLE, STRAIGHT DIVISOR WITH 90 AS THE HIGH-ORDER BYTE IS ADDED TO THE DIVIDEND. THE 90 IS IN ALIGNMENT WITH THE HIGH-ORDER DIVIDEND BYTE. THE ADD CYCLES CONTINUE UNTIL DIGIT TO THE RIGHT OF THE 9 GOES TO ZERO.
3. SKEWED DIVISOR WITH 09 AS THE HIGH-ORDER BYTE IS SUBTRACTED FROM THE PREVIOUS RESULT WITH ALIGNMENT SHIFTED ONE DIGIT. SUBTRACT CYCLES CONTINUE UNTIL DIGIT TO THE RIGHT OF THE 9 GOES TO NINE
4. ALTERNATE STRAIGHT AND SKEWED CYCLES ARE TAKEN WITH DIVISOR SHIFTING ONE DIGIT FOR EACH CHANGE. OPERATION ENDS WHEN THE CYCLE THAT INCLUDES THE LEFTMOST POSITION OF THE DIVIDEND IS COMPLETE.
5. QUOTIENT SIGN IS INSERTED INTO LOW-ORDER 4 BITS OF QUOTIENT. SIGN OF REMAINDER IS INSERTED INTO LOW-ORDER 4 BITS OF REMAINDER.

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                 CDMD 001    T                    DECIMAL MULTIPLY DIVIDE ROUTINES
17A8     E1AD    CDMD 002     SSOPFC      004     NOFILC     BR IF S6=1
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
17AA  1445  CDMD 003                                     G0=G0*-K40         FIX ILC FOR EXECUTE
17AC  C130  CDMD 004      NOFILC       006   LENCHK     BR IF G14=0           LENGTH CHECK OVER 7
17AE  AE94  CDMD 005      FLGERR  CICY 070   SPECHK     BR
17B0  5575  CDMD 006      LENCHK                        D1=G1XL            CHECK IF
17B2  77E3  CDMD 007                                    D1=D1-H0            L2 LARGER
17B4  E0AE  CDMD 008                 005   FLGERR     BR IF HZNZ                THAN L1
17B6  9812  CDMD 009            CCOM 056   LSAVE      BAL
17B8  5730  CDMD 010                                    RDB D1 V           READ 2ND OPRAND
17BA  8F2C  CDMD 011                 203   SINSET     BAL                 GO SET THE SIGN VALUE
17BC  5710  CDMD 012                                    RDB D1 U           READ 1ST OPRAND
17BE  3415  CDMD 013                                    G0=G0$K10           START WITH REMAINDER SIGN +
17C0  8F2C  CDMD 014                 203   SINSET     BAL                 GO SET THE SIGN VALUE
17C2  10CE  CDMD 015                                    RST S K=FC         RESET S REG 0,1,2,3,4,5
17C4  55BD  CDMD 016                                    T1=G1L             -BUILD THE
17C6  3B83  CDMD 017                                    T1=T1$K08            STRAIGHT AND SKEWED ADDRESSES
17C8  3B15  CDMD 018                                    T1=T1$K10            AND PUT THEM IN AUX
17CA  7AE2  CDMD 019                                    STH T DA,BC            OOBC-STRAIGHT
17CC  4CA6  CDMD 020                                    P=T                    OOEF-SKEWED
17CE  3B25  CDMD 021                                    T1=T1$K20
17D0  7AF2  CDMD 022                                    STH T DA,BE
17D2  5F3A  CDMD 023                                    RDB H1 V-1         -THIS STORES THE DIVISOR
17D4  5FFB  CDMD 024                                    H1=H1H               PLIER IN AUX STRAIGHT AND
17D6  7FAF  CDMD 025                                    H1C=H1@T0+C           SKEWED
17D8  7FCA  CDMD 026                                    STB H1 AS,P-1
17DA  5FF1  CDMD 027                                    H1=H1X
17DC  C96C  CDMD 028                 036   SSKEW      BR IF P14=0        TAKE THIS BR IF 1 POS. FIELD
17DE  573A  CDMD 029      SMORE                         RDB D1 V-1
17E0  47F3  CDMD 030                                    H1=D1XH+H1L
17E2  77AF  CDMD 031                                    D1C=D1@T0+C
17E4  7FAA  CDMD 032                                    STB H1 AS,T-1
17E6  77CA  CDMD 033                                    STB D1 AS,P-1
17E8  57F5  CDMD 034                                    H1=D1XL
17EA  C95F  CDMD 035                 029   SMORE      BR IF P14=1
17EC  7FAA  CDMD 036      SSKEW                         STB H1 AS,T-1        STORE LAST SKEWED CHAR.
17EE  55E5  CDMD 037                                    H0=G1XL            -THIS ADJUSTS THE ADDRESS 1ST OP
17F0  55FD  CDMD 038                                    H1=G1L              TO THE LEFT MOST POS.
17F2  7FF1  CDMD 039                                    H0=H0-H1+1
17F4  2002  CDMD 040                                    SET S3
17F6  71E9  CDMD 041                                    U1C=U1-H0+C
17F8  70A9  CDMD 042                                    U0C=U0-T0+C
17FA  4206  CDMD 043                                    V=U
            CDMD 044  *                                                    BREAK OUT
17FC  F027  CDMD 045                 047   SOUT       BR IF G07=1         THE OP
17FE  8D82  CDMD 046                 106   MULTPY     BR                   CODES
17A6  A51C  CDMD 047      SOUT       052   DIVIDE     BR
            CDMD 048  *              AT THIS POINT THE DIVIDE OPERATION
            CDMD 049  *               IS STARTED IT GOES BACK TO A COMMON
            CDMD 050  *                POINT WITH MULTIPLY TO SET THE RESULT SIGN
            CDMD 051  *
251C  B3C8  CDMD 052      DIVIDE     213   DVLOOP     BAL                GO SEE IF ANS. WILL FIT
251E  C1A4  CDMD 053                 056   VALIDA     BR IF S4=0
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
2520  8174  CDMD 054      DATEXP   CCOM 064  LRSTR      BAL                THIS RESTORES LOCAL STORAGE AND
2522  98DC  CDMD 055               CCOM 071  FLGDEX     BR                  FLAGS A DATA EXCEPTION INTERUPT
2524  F5B0  CDMD 056      VALIDA        062  VALIFL     BR IF S3=0         IF S3=1 A DIVIDE EXCEPTION EXIST
2526  1002  CDMD 057                                   RST S K=90          ADD BACK IN
2528  B3CA  CDMD 058                    214  DVLOPB     BAL                 THE VALUE
252A  8174  CDMD 059               CCOM 064  LRSTR      BAL                  SUB
252C  2BB3  CDMD 060                                   T1=0$K0B              AND FLAG
252E  89FC  CDMD 061               BPSW 003  PROG       BR                     DIV. EXCEP. INTER.
2530  56A2  CDMD 062      VALIFL                        RDH D DA,AC        GET
2532  5D70  CDMD 063                                   RDB P1 D            RID
2534  5DDB  CDMD 064                                   P1=P1H               OF
2536  7D70  CDMD 065                                   STB P1 D              SIGN
2538  2993  CDMD 066                                   I1=0$K09
253A  5981  CDMD 067                                   I0=I1X
253C  5004  CDMD 068      ENTYA                         U=U+1
253E  1042  CDMD 069      ENTYB                         RST S K=94         RESET S REG  0,3,5
2540  56E2  CDMD 070                                   RDH D DA,BC         GET THE STRAIGHT ADDRESS
2542  B3CC  CDMD 071                    215  DVLOPC     BAL
2544  5D3C  CDMD 072                                   RDB P1 V
2546  7D8F  CDMD 073                                   P1C=P1@I0+C
2548  F081  CDMD 074                    077  OTHLOP     BR IF LZ=0
254A  7D30  CDMD 075                                   STB P1 V
254C  A53E  CDMD 076                    069  ENTYB      BR
2500  7D30  CDMD 077      OTHLOP                        STB P1 V
2502  7EA3  CDMD 078                                   H0=H0-T0
2504  C4CF  CDMD 079                    091  CHECK      BR IF Z=0
2506  B3C8  CDMD 080      OTALOP        213  DVLOOP     BAL
2508  0D9D  CDMD 081                                   Z=P1<K90
250A  F08E  CDMD 082                    084  ARDN       BR IF HZNZ
250C  2040  CDMD 083                                   SET S5
250E  5D30  CDMD 084      ARDN                          RDB P1 V
2510  5D7B  CDMD 085                                   D1=P1H
2512  7D9F  CDMD 086                                   P1C=P1@I1+C
2514  47DB  CDMD 087                                   P1=D1H+P1L
2516  7D30  CDMD 088                                   STB P1 V
2518  D186  CDMD 089                    080  OTALOP     BR IF S5=0
251A  A53C  CDMD 090                    068  ENTYA      BR
254E  C1A1  CDMD 091      CHECK         054  DATEXP     BR IF S4=1
2550  5CC2  CDMD 092                                   RDH P DA,B8
2552  2FC3  CDMD 093                                   H1=0$K0C
2554  C958  CDMD 094                    096  ARDNB      BR IF P14=0
2556  1F6B  CDMD 095                                   H1=H1<K06
2558  E45D  CDMD 096      ARDNB         098  ARDNC      BR IF G02=1          THIS
255A  3F13  CDMD 097                                   H1=H1$K01             SETS
255C  5730  CDMD 098      ARDNC                         RDB D1 V               THE
255E  67F5  CDMD 099                                   D1=D1$H1                QUO
2560  7730  CDMD 100                                   STB D1 V                 SIGN
2562  6443  CDMD 101                                   G0=G0+G0
2564  8DC2  CDMD 102                    138  MEND       BR
            CDMD 103    *
            CDMD 104    *
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CDMD 105    *
0D82   571A   CDMD 106     MULTPY                          RDB D1 U-1         CHECK
0D84   A31B   CDMD 107                    150   EREXCP N   BR IF D1=NZ         FIELD FOR
0D86   2FFF   CDMD 108                                     H1=H1+KFF            ENOUGH ZEROS
0D88   F483   CDMD 109                    106   MULTPY     BR IF AC=1
0D8A   5004   CDMD 110                                     U=U+1
0D8C   5224   CDMD 111                                     V=V+1
0D8E   5EF9   CDMD 112                                     H1=H0
0D90   2807   CDMD 113                                     T1=0
0D92   E597   CDMD 114                    116   MNZERO     BR IF S2=1
0D94   2040   CDMD 115                                     SET S5
0D96   0004   CDMD 116     MNZERO                          RST S2
0D98   77AF   CDMD 117     MLUPA                           D1C=D1@T0+C        STORE THE
0D9A   5730   CDMD 118                                     RDB D1 V             1ST
0D9C   D1A1   CDMD 119                    121   MZER       BR IF S5=1             OP
0D9E   7718   CDMD 120                                     STB D1 U+1               SHIFTED
0DA0   7B38   CDMD 121     MZER                            STB T1 V+1                 TO
0DA2   2FFF   CDMD 122                                     H1=H1+KFF                    LEFT
0DA4   C498   CDMD 123                    117   MLUPA      BR IF ZNZ
0DA6   5006   CDMD 124                                     U=U-1
0DA8   5226   CDMD 125                                     V=V-1
0DAA   D1C3   CDMD 126                    138   MEND       BR IF S5=1
0DAC   577B   CDMD 127                                     D1=D1H             RID OG
0DAE   7710   CDMD 128                                     STB D1 U             SIGN
0DB0   77AF   CDMD 129                                     D1C=D1@T0+C
0DB2   C181   CDMD 130                    150   EREXCP 0   BR IF S4=1
0DB4   E5C2   CDMD 131                    138   MEND       BR IF S2=0
0DB6   B12E   CDMD 132                    151   MSTART     BR
0DB8   0B9D   CDMD 133     ENDING                          Z=T1<K90
0DBA   E0C2   CDMD 134                    138   MEND       BR IF HZNZ
0DBC   4086   CDMD 135                                     U=I
0DBE   1002   CDMD 136                                     RST S K=90
0DC0   B3CA   CDMD 137                    214   DVLOPB     BAL
0DC2   54F9   CDMD 138     MEND                            H1=G0              SET THE
0DC4   8174   CDMD 139                CCOM 064   LRSTR      BAL                  SIGN
0DC6   27C3   CDMD 140                                     D1=0$KCC               FOR
0DC8   C94C   CDMD 141                    143   MASCI      BR IF P14=0              REMAIN
0DCA   176B   CDMD 142                                     D1=D1<K06                  AND
0DCC   EF51   CDMD 143     MASCI          145   MPLUS      BR IF H12=1                PRODUCT
0DCE   3713   CDMD 144                                     D1=D1$K01
0DD0   5F10   CDMD 145     MPLUS                           RDB H1 U
0DD2   6F75   CDMD 146                                     H1=H1$D1
0DD4   7F10   CDMD 147     STORE                           STB H1 U
0DD6   2A07   CDMD 148     TOTOO                           T0=0
0DD8   AA76   CDMD 149                CICY 004   CHECK      BR
0D80   A520   CDMD 150     EREXCP 0       054   DATEXP     BR
312E   2040   CDMD 151     MSTART                          SET S5
3130   4826   CDMD 152                                     I=V                THIS
3132   5789   CDMD 153     MSB                             T1=D1                DOES
3134   C5B8   CDMD 154     MSJKA          156   MSA        BR IF S0=0             THE
3136   0B9F   CDMD 155                                     Z=T1<K99               ACTUAL
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
3138   008E    CDMD 156    MSA                          RST S K=78          MULTIPLY
313A   D1CC    CDMD 157                 174   MSLOW      BR IF S5=0           S0 1-SUB
313C   E0E0    CDMD 158    MSHI         166   MSC        BR IF HZNZ              0-ADD
313E   7B1A    CDMD 159    MSSTOR                       STB T1 U-1           S5 0-LOW HALF
3140   5886    CDMD 160                                 I=I-1                   1-HI HALF
3142   5710    CDMD 161                                 RDB D1 U
3144   0040    CDMD 162                                 RST S5              H0=NUMBER OF BYTES
3146   2EFF    CDMD 163                                 H0=H0+KFF
3148   C4B2    CDMD 164                 153   MSB        BR IF ZNZ
314A   8DB8    CDMD 165                 133   ENDING     BR
3160   2040    CDMD 166    MSC                          SET S5
3162   2A95    CDMD 167                                 T0=0$K90
3164   56E2    CDMD 168                                 RDH D DA,BC
3166   57C9    CDMD 169                                 P0=D1
3168   1000    CDMD 170                                 RST S0
316A   0BB9    CDMD 171                                 Z=T1+KB0
316C   F4DD    CDMD 172                 182   SUBCYC     BR IF AC=1
316E   9456    CDMD 173                 185   CALLOP     BR
314C   F0BD    CDMD 174    MSLOW        158   MSHI       BR IF LZ=0
314E   5BBD    CDMD 175                                 T1=T1L
3150   2A93    CDMD 176                                 T0=0$K09
3152   56F2    CDMD 177                                 RDH D DA,BE
3154   57C9    CDMD 178                                 P0=D1
3156   1000    CDMD 179                                 RST S0
3158   0BB1    CDMD 180                                 Z=T1+K0B
315A   E0DF    CDMD 181                 183   ADDCYC     BR IF HZ=0
315C   3002    CDMD 182    SUBCYC                       SET S K=90
315E   9456    CDMD 183    ADDCYC       185   CALLOP     BR
1454   5C79    CDMD 184    MSJK                         D1=P0
1456   4286    CDMD 185    CALLOP                       V=I
1458   5D30    CDMD 186    MSCLOP                       RDB P1 V            CALCULATE
145A   5F6A    CDMD 187                                 RDB H1 AS,D-1         LOOP
145C   7DFF    CDMD 188                                 P1C=P1@H1+C
145E   7D3A    CDMD 189                                 STB P1 V-1
1460   C359    CDMD 190                 186   MSCLOP     BR IF D14=1
1462   7BAF    CDMD 191                                 T1C=T1@T0+C
1464   0002    CDMD 192                                 RST S3
1466   C5EC    CDMD 193                 196   MSJA       BR IF S0=0
1468   0B9F    CDMD 194                                 Z=T1<K99
146A   2002    CDMD 195                                 SET S3
146C   D1F2    CDMD 196    MSJA         199   MSKY       BR IF S5=0
146E   E0D4    CDMD 197                 184   MSJK       BR IF HZNZ
1470   B13E    CDMD 198                 159   MSSTOR     BR
1472   F0D4    CDMD 199    MSKY         184   MSJK       BR IF LZNZ
1474   5710    CDMD 200                                 RDB D1 U
1476   47BB    CDMD 201                                 T1=D1H+T1L
1478   B134    CDMD 202                 154   MSJKA      BR
0F2C   5773    CDMD 203    SINSET                       D1=D1XH                   SIGN CHECK AND SET ROU
0F2E   0769    CDMD 204                                 Z=D1+K60                GO IS USED TO PUT THE SIGNS
0F30   F4B5    CDMD 205                 207   SIGVAL     BR IF AC=1
0F32   A520    CDMD 206                 054   DATEXP     BR                  BRANCH TO DATA EXCEPTION
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS

0F34    F73C    CDMD 207    SIGVAL       211  EXIT       BR IF D13=0          1010- PLUS     1101-MINUS
0F36    0719    CDMD 208                                  Z=D1+K10             1011- MINUS    1110-PLUS
0F38    F48D    CDMD 209                 211  EXIT       BR IF AC=1           1100- PLUS     1111-PLUS
0F3A    143D    CDMD 210                                  G0=G0<K30               GO BIT 2  0=ANS MINUS 1-PLS
0F3C    128E    CDMD 211    EXIT                          RTN                     GO BOT 3  0=REM NEG  1-PLUS
                CDMD 212    *
33C8    3002    CDMD 213    DVLOOP                        SET S K=90           SET S 0,3  FOR SUB CYCLE
33CA    56F2    CDMD 214    DVLOPB                        RDH D DA,BE         GET THE SKEWED ADDRESS
33CC    4206    CDMD 215    DVLOPC                        V=U                  SET UP V FOR LOOP
33CE    5D30    CDMD 216    ALOOP                         RDB P1 V            ACTUAL LOOP
33D0    5F6A    CDMD 217                                  RDB H1 AS,D-1        FOR DOING
33D2    7DFF    CDMD 218                                  P1C=P1@H1+C           THE ADD OR
33D4    7D3A    CDMD 219                                  STB P1 V-1             SUB FUNCTION
33D6    C34F    CDMD 220                 216  ALOOP      BR IF D14=1
33D8    128E    CDMD 221                                  RTN
                                                 **********************************
                                                 * CROSS REFERENCE FOR CSECT CDMD *
                                                 **********************************

CDMD 002     CICY 284  CICY 285
CDMD 004     CDMD 002
CDMD 005     CDMD 008
CDMD 006     CDMD 004
CDMD 029     CDMD 035
CDMD 036     CDMD 028
CDMD 047     CDMD 045
CDMD 052     CDMD 047
CDMD 054     CDMD 091  CDMD 150  CDMD 206
CDMD 056     CDMD 053
CDMD 062     CDMD 056
CDMD 068     CDMD 090
CDMD 069     CDMD 076
CDMD 077     CDMD 074
CDMD 080     CDMD 089
CDMD 084     CDMD 082
CDMD 091     CDMD 079
CDMD 096     CDMD 094
CDMD 098     CDMD 096
CDMD 106     CDMD 046  CDMD 109
CDMD 116     CDMD 114
CDMD 117     CDMD 123
CDMD 121     CDMD 119
CDMD 133     CDMD 165
CDMD 138     CDMD 102  CDMD 126  CDMD 131  CDMD 134
CDMD 143     CDMD 141
CDMD 145     CDMD 143
CDMD 150     CDMD 107  CDMD 130
CDMD 151     CDMD 132
CDMD 153     CDMD 164
CDMD 154     CDMD 202
CDMD 156     CDMD 154
CDMD 158     CDMD 174
```

****
* CROSS REFERENCE FOR CSECT CDMD *
****

CDMD 159	CDMD 198		
CDMD 166	CDMD 158		
CDMD 174	CDMD 157		
CDMD 182	CDMD 172		
CDMD 183	CDMD 181		
CDMD 184	CDMD 197	CDMD 199	
CDMD 185	CDMD 173	CDMD 183	
CDMD 186	CDMD 190		
CDMD 196	CDMD 193		
CDMD 199	CDMD 196		
CDMD 203	CDMD 011	CDMD 014	CNVR 070
CDMD 207	CDMD 205		
CDMD 211	CDMD 207	CDMD 209	
CDMD 213	CDMD 052	CDMD 080	
CDMD 214	CDMD 058	CDMD 137	
CDMD 215	CDMD 071		
CDMD 216	CDMD 220		

CDVD DESCRIPTIVE TEXT

ENTRY	OPERATION	BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP1D	BINARY DIVIDE	1D	R1 R2		
RXOP5D	BINARY DIVIDE	5D	R1 X2	B2 D2	D2

OBJECTIVES

THE DIVIDEND (1ST OPERAND) IS DIVIDED BY THE DIVISOR (2ND OPERAND). THE QUOTIENT AND REMAINDER ARE PLACED IN THE FIRST OPERAND LOCATION.

THE DIVIDEND IS A 64 BIT SIGNED INTEGER THAT MUST BE LOCATED IN AN EVEN-ODD PAIR OF GENERAL PURPOSE REGISTERS. THE DIVISOR IS A 32 BIT SIGNED INTEGER. THE QUOTIENT SIGN IS DETERMINED BY THE RULES OF ALGEBRA.

EXECUTION

READOUT THE DIVISOR
CHECK FOR EVEN DIVIDEND ADDRESS
IF ODD ADDRESS, FLAG SPECIFICATION EXCEPTION IN CICY ROUTINE AND GO STORE INTERRUPT CODE IN PSW.
READOUT THE DIVIDEND
IF DIVIDEND MINUS, COMPLEMENT IT.
IF DIVISOR MINUS, COMPLEMENT IT.
DO A TRIAL SUBTRACT
IF QUOTIENT WILL FIT, DO THE DIVIDE.
IF QUOTIENT TOO LARGE, FLAG A DIVIDE EXCEPTION AND GO TO STORE INTERRUPT CODE IN PSW.
DO THE SUBTRACT
THE QUOTIENT AND REMAINDER ARE PLACED IN THE FIRST OPERAND LOCATION.

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

            CDVD 001   T                    BINARY DIVIDE ROUTINES
0390  E195  CDVD 002    RROP1D             004  NEQLAB    BR IF S6=1
0392  3485  CDVD 003                       GO=G0$K80
0394  8AC8  CDVD 004    NEQLAB   CCOM 029  LGET         BAL
0396  B2C2  CDVD 005                  009  MEET         BR
32BC  5618  CDVD 006    RXOP5D             RDH D U+2              READ
32BE  5E10  CDVD 007                       RDH H U                 OUT DIV
32C0  55BB  CDVD 008                       T1=G1H
32C2  F546  CDVD 009    MEET          011  MEETA        BR IF G13=0
32C4  AE94  CDVD 010             CICY 070  SPECHK       BR
32C6  9814  CDVD 011    MEETA    CCOM 057  LSAVEA       BAL
32C8  54A8  CDVD 012                       RDH G AS,T+2           READ
32CA  58AA  CDVD 013                       RDH I AS,T-2             OUT
32CC  3B15  CDVD 014                       T1=T1$K10                 THE
32CE  52A8  CDVD 015                       RDH V AS,T+2               DVD
32D0  50AA  CDVD 016                       RDH U AS,T-2
32D2  C658  CDVD 017                  020  MEETB        BR IF D00=0
32D4  B3DA  CDVD 018             CCOM 076  COMPL        BAL                    MINUS
32D6  3B13  CDVD 019                       T1=T1$K01               DIV
32D8  C45E  CDVD 020    MEETB          023  MEETD        BR IF G00=0
32DA  9676  CDVD 021                  088  COMDVD       BAL                    MINUS
32DC  3B23  CDVD 022                       T1=T1$K02               DVD
32DE  4C86  CDVD 023    MEETD              P=I                    SETUP
32E0  3002  CDVD 024                       SET S K=90              AND DO
32E2  7DFD  CDVD 025                       P1C=P1%H1+C               TRIAL
32E4  7CFD  CDVD 026                       P0C=P0%H0+C                  SUBTRACT
32E6  4C46  CDVD 027                       P=G
32E8  7D7D  CDVD 028                       P1C=P1%D1+C
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
32EA    7C6D    CDVD 029                                    P0C=P0%D0+C
32EC    F5F0    CDVD 030                  032    MEETC      BR IF S3=0             BRANCJ OK
32EE    ABFC    CDVD 031                  098    DIVCHK     BR
32F0    4C52    CDVD 032    MEETC                            RDH P DC,9A
32F2    10CE    CDVD 033                                     RST S K=FC
32F4    AEF8    CDVD 034                  047    SHIFTA     BAL
32F6    8DDC    CDVD 035                  037    ARND       BR
0DDA    AEF6    CDVD 036    INSERT        046    SHIFT      BAL
0DDC    3002    CDVD 037    ARND                             SET S K=90
0DDE    79FD    CDVD 038    COMPUT                           I1C=I1%H1+C
0DE0    78ED    CDVD 039                                     I0C=I0%H0+C
0DE2    757D    CDVD 040                                     G1C=G1%D1+C
0DE4    746D    CDVD 041                                     G0C=G0%D0+C
0DE6    F5DB    CDVD 042                  036    INSERT     BR IF S3=1
0DE8    AEF6    CDVD 043                  046    SHIFT      BAL
0DEA    1002    CDVD 044                                     RST S K=90
0DEC    8DDF    CDVD 045                  038    COMPUT     BR
2EF6    2C1B    CDVD 046    SHIFT                            P0=P0+K01
2EF8    611D    CDVD 047    SHIFTA                           U1C=U1+U1+C            SHIFT
2EFA    600D    CDVD 048                                     U0C=U0+U0+C             LEFT
2EFC    633D    CDVD 049                                     V1C=V1+V1+C              ROUTINE
2EFE    622D    CDVD 050                                     V0C=V0+V0+C
2F00    EC0D    CDVD 051                  057    CYCLE      BR IF P02=1
2F02    699D    CDVD 052                                     I1C=I1+I1+C
2F04    688D    CDVD 053                                     I0C=I0+I0+C
2F06    655D    CDVD 054                                     G1C=G1+G1+C
2F08    644D    CDVD 055                                     G0C=G0+G0+C
2F0A    128E    CDVD 056                                     RTN
2F0C    C418    CDVD 057    CYCLE         063    ARNQ       BR IF G00=0
2F0E    1002    CDVD 058                                     RST S K=90
2F10    79FD    CDVD 059                                     I1C=I1%H1+C
2F12    78ED    CDVD 060                                     I0C=I0%H0+C
2F14    757D    CDVD 061                                     G1C=G1%D1+C
2F16    746D    CDVD 062                                     G0C=G0%D0+C
2F18    5BC9    CDVD 063    ARNQ                             P0=T1
2F1A    52D9    CDVD 064                                     P1=V0
2F1C    E827    CDVD 065                  070    NXTA       BR IF P06=1            REM -
2F1E    C436    CDVD 066                  078    NXTB       BR IF G00=0            REM SHD BE +   IS +
2F20    968A    CDVD 067                  100    COMR       BAL
2F22    C436    CDVD 068                  078    NXTB       BR IF G00=0
2F24    ABFC    CDVD 069    LABELQ        098    DIVCHK     BR
2F26    C42B    CDVD 070    NXTA          072    NXTD       BR IF G00=1
2F28    968A    CDVD 071                  100    COMR       BAL
2F2A    F839    CDVD 072    NXTD          079    NXTF       BR IF P07=1
2F2C    968E    CDVD 073    NXTC          102    COMQUO     BAL
2F2E    52D9    CDVD 074                                     P1=V0
2F30    D58A    CDVD 075                  080    FINISH     BR IF S1=0
2F32    CD3B    CDVD 076                  080    FINISH     BR IF P1 BIT0=1
2F34    ABFC    CDVD 077                  098    DIVCHK     BR
2F36    F82D    CDVD 078    NXTB          073    NXTC       BR IF P07=1
2F38    CD25    CDVD 079    NXTF          069    LABELQ     BR IF P10=1
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
2F3A    5BBB    CDVD 080    FINISH                          T1=T1H
2F3C    72A8    CDVD 081                                    STH V AS,T+2
2F3E    70AA    CDVD 082                                    STH U AS,T-2
2F40    1B15    CDVD 083                                    T1=T1*-K10
2F42    74A8    CDVD 084                                    STH G AS,T+2
2F44    78AA    CDVD 085                                    STH I AS,T-2
2F46    8176    CDVD 086            CCOM 065  LRSTRA        BAL
2F48    AA76    CDVD 087            CICY 004  CHECK         BR
1676    2002    CDVD 088    COMDVD                          SET S K=10         COMP
1678    711B    CDVD 089                                    U1C=C-U1+C          DVD
167A    700B    CDVD 090                                    U0C=0-U0+C
167C    733B    CDVD 091                                    V1C=0-V1+C
167E    7228    CDVD 092                                    V0C=0-V0+C
1680    799B    CDVD 093    CMDVA                           I1C=C-I1+C
1682    788B    CDVD 094                                    I0C=0-I0+C
1684    755B    CDVD 095                                    G1C=0-G1+C
1686    744B    CDVD 096                                    G0C=0-G0+C
1688    128E    CDVD 097                                    RTN
2BFC    8176    CDVD 098    DIVCHK    CCOM 065  LRSTRA      BAL
2BFE    ABE2    CDVD 099              CNVR 120  LAB14       BR                 GO FLAG DIVIDE CHECK AND INTRPT
168A    2002    CDVD 100    COMR                            SET S K=10
168C    F581    CDVD 101                 093    CMDVA       BR IF S3=1
168E    0008    CDVD 102    COMQUO                          RST S1
1690    3002    CDVD 103                                    SET S K=90
1692    711B    CDVD 104                                    U1C=0-U1+C
1694    700B    CDVD 105                                    U0C=0-U0+C
1696    733B    CDVD 106                                    V1C=0-V1+C
1698    722B    CDVD 107                                    V0C=0-V0+C
169A    128E    CDVD 108                                    RTN
```

```
**********************************
* CROSS REFERENCE FOR CSECT CDVD *
**********************************
```

```
CDVD 002    CICY 182
CDVD 004    CDVD 002
CDVD 006    CICY 214
CDVD 009    CDVD 005
CDVD 011    CDVD 009
CDVD 020    CDVD 017
CDVD 023    CDVD 020
CDVD 032    CDVD 030
CDVD 036    CDVD 042
CDVD 037    CDVD 035
CDVD 038    CDVD 045
CDVD 046    CDVD 036  CDVD 043
CDVD 047    CDVD 034
CDVD 057    CDVD 051
CDVD 063    CDVD 057
CDVD 069    CDVD 079
CDVD 070    CDVD 065
CDVD 072    CDVD 070
CDVD 073    CDVD 078
```

****
* CROSS REFERENCE FOR CSECT CDVD *
****

CDVD 078 CDVD 066 CDVD 068
CDVD 079 CDVD 072
CDVD 080 CDVD 075 CDVD 076
CDVD 088 CDVD 021
CDVD 093 CDVD 101
CDVD 098 CDVD 031 CDVD 069 CDVD 077
CDVD 100 CDVD 067 CDVD 071
CDVD 102 CDVD 073

CFAD DESCRIPTIVE TEXT

ENTRY IS MADE TO THE -CFAD- ROUTINE FROM FLOATING POINT I-CYCLES -CFCY-. THE NORMALIZED AND UNNORMALIZED ADD OR SUBTRACT, AND COMPARE OP CODES LISTED BELOW ARE HANDLED BY THIS ROUTINE.

OP CODE TYPE	BYTE 1	BYTE 2	BYTE 3	BYTE 4
RR COMPARE (LONG)	29	R1 R2		
RR ADD NORM. (LONG)	2A	R1 R2		
RR SUB. NORM. (LONG)	2B	R1 R2		
RR ADD UNORM. (LONG)	2E	R1 R2		
RR SUB UNORM. (LONG)	2F	R1 R2		
RR COMPARE (SHORT)	39	R1 R2		
RR ADD NORM. (SHORT)	3A	R1 R2		
RR SUB NORM. (SHORT)	3B	R1 R2		
RR ADD UNORM. (SHORT)	3E	R1 R2		
RR SUB UNORM. (SHORT)	3F	R1 R2		
RX COMPARE (LONG)	69	R1 X2	B2 D2	D2
RX ADD NORM. (LONG)	6A	R1 X2	B2 D2	D2
RX SUB NORM. (LONG)	6B	R1 X2	B2 D2	D2
RX ADD UNORM. (LONG)	6E	R1 X2	B2 D2	D2
RX SUB UNORM. (LONG)	6F	R1 X2	B2 D2	D2
RX COMPARE (SHORT)	79	R1 X2	B2 D2	D2
RX ADD NORM. (SHORT)	7A	R2 X2	B2 D2	D2
RX SUB NORM. (SHORT)	7B	R2 X2	B2 D2	D2
RX ADD UNORM. (SHORT)	7E	R1 X2	B2 D2	D2
RX SUB UNORM. (SHORT)	7F	R1 X2	B2 D2	D2

OBJECTIVES

COMPARE OPS - THE FIRST OPERAND IS COMPARED WITH THE SECOND OPERAND. THE CONDITION CODE IS SET TO INDICATE THE RESULT OF THE COMPARE. COMPARISON TAKES INTO ACCOUNT THE SIGN, FRACTION, AND EXPONENT OF EACH OPERAND.

ADD OPS - THE SECOND OPERAND IS ADDED TO THE FIRST OPERAND, AND THE SUM IS PLACED IN THE FIRST OPERAND LOCATION. FOR SHORT OP FORMATS, THE LOW ORDER HALVES OF THE FLOATING POINT REGISTERS ARE NOT USED.
CHARACTERISTICS ARE ALIGNED BEFORE ADDITION TAKES PLACE.

SUBTRACT OPS- THE SECOND OPERAND IS SUBTRACTED FROM THE FIRST OPERAND, AND THE DIFFERENCE IS PLACED IN THE FIRST OPERAND LOCATION. FOR SHORT OP FORMATS THE LOW ORDER HALVES OF THE FLOATING POINT REGISTERS ARE NOT USED.
CHARACTERISTICS ARE ALIGNED BEFORE SUBTRACTION TAKES PLACE.

CONDITION CODES

COMPARE	0=OPERANDS ARE EQUAL 1=FIRST OPERAND IS LOW 2=FIRST OPERAND IS HIGH
ADD AND SUBTRACT	0=RESULT FRACTION IS ZERO 1=RESULT IS LESS THAN ZERO 2=RESULT IS GREATER THAN ZERO 3=RESULT EXPONENT OVERFLOWS

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT              COMMENTS
                CFAD 001    T                                  ADD,SUBTRACT,COMPARE   FLOATING POINT
                CFAD 002    *
                CFAD 003    *
                CFAD 004    *
                CFAD 005    *
                CFAD 006    ASEQ    AL07=72
0AF2    3B35    CFAD 007     SCAD                              T1=T1$K30 ** T=ADD OF SAVE AREA-38 OR 78 IN AUX
0AF4    896A    CFAD 008              CFCO 078   STORE2       BAL                    SAVE 2ND OPERAND
0AF6    5E79    CFAD 009                                       D1=H0                  SAVE R2 EXPONENT
0AF8    5AE2    CFAD 010                                       RDH T DA,BC            GET R1 ADDRESS
0AFA    B42E    CFAD 011              CFCO 090   RDOPER       BAL                    GET R1 OPERAND
0AFC    2C07    CFAD 012                                       P0=0
0AFE    5E69    CFAD 013                                       D0=H0                  SAVE R1 EXP
0B00    5ED9    CFAD 014                                       P1=H0                  SAVE R1 EXP FOR ORIG SIGN
0B02    2040    CFAD 015                                       SET S5  ** S5  WILL BE RST IF R1 MUST BE RIGHT SHIFT
0B04    7671    CFAD 016                                       D0=D0-D1+1             SUBTRACT R1EXP-R2EXP
                CFAD 017    *                                                NO 1BC IND. R1 MUST SHIFT  D0 HAS COUNT
0B06    E48E    CFAD 018                   022   NOTOVF       BR IF NOVFL
0B08    F491    CFAD 019                   023   NOT1BC       BR IF AC=1             OVFL IS ON
0B0A    56C9    CFAD 020     HAV1BC                           P0=D0  ** SHIFT COUNT TRUE FORM TO P0
0B0C    8816    CFAD 021                   027   TCCTRL       BR
0B0E    F48B    CFAD 022     NOTOVF        020   HAV1BC       BR IF AC=1
0B10    0040    CFAD 023     NOT1BC                           RST S5 ** NO 1BC R1 IS SMALL-MUST  SHIFT
                CFAD 024    *                                          R2 IS RES EXP MOVE R2 EXP TO R1 OPERAND
0B12    57E9    CFAD 025                                       H0=D1
0B14    7C61    CFAD 026                                       P0=P0-D0+1 ** SHIFT COUNT IS IN COMP. MUST RECOMP
0B16    F01A    CFAD 027    TCCTRL         029   THISAD       BR IF G07=0            BR IF ADD COMP CTRL IS OK
0B18    1C8D    CFAD 028                                       P0=P0<K80              SUB,COMP-MUST INVERT T/C BIT
0B1A    CC1E    CFAD 029    THISAD         031   TRUE         BR IF P00=0            BR IF TRUE
0B1C    3002    CFAD 030                                       SET S K=90   ** SET  COMP CTRL AND CARRY
0B1E    1C85    CFAD 031    TRUE                               P0=P0*-K80             REMOVE SIGN FROM SHIFT COUNT
0B20    E0A5    CFAD 032                   034   SPEEDY       BR IF HZ=0             BR IF NO HI COUNT ALIGNMENT
0B22    2CF3    CFAD 033                                       P0=0$K0F               USE COUNT OF 15 TO ZERO OUT REG
0B24    F0BF    CFAD 034    SPEEDY         047   NOSHIF       BR IF LZ=0             BR IF EXPONENTS ARE EQUAL
0B26    D1B6    CFAD 035                   043   SHIFR1       BR IF S5=0             BR IF R1 MUST SHIFT
0B28    3B35    CFAD 036                                       T1=T1$K30              T= R2 SAVE
0B2A    B42E    CFAD 037              CFCO 090   RDOPER       BAL                    GET R2
0B2C    A656    CFAD 038              CFCO 007   SHIFR4       BAL
0B2E    896A    CFAD 039              CFCO 078   STORE2       BAL                    PUT SHIFTED R2 BACK IN SAVE
0B30    5AE2    CFAD 040                                       RDH T DA,BC            GET R1 ADD  T=C0X8
0B32    B42E    CFAD 041              CFCO 090   RDOPER       BAL
0B34    8B40    CFAD 042                   048   ADD          BR
0B36    1E85    CFAD 043    SHIFR1                             H0=H0*-K80             MAKE RESULT SIGN +
0B38    CD3C    CFAD 044                   046   DESADD       BR IF P10=0            BR IF ORIG SIGN + SIGN IS OK
0B3A    3E85    CFAD 045                                       H0=H0$K80              ORIG WAS - MAKE RES -
0B3C    A656    CFAD 046    DESADD    CFCO 007   SHIFR4       BAL                    GO SHIFT R1 RIGHT
0B3E    0040    CFAD 047    NOSHIF                             RST S5  **  NO SHIFT RST S5 FOR GUARD DIGIT CTRL
0B40    3B35    CFAD 048    ADD                                T1=T1$K30              T= 38 OR 78 =R2 SAVE
0B42    0008    CFAD 049                                       RST S1                 FOR GUARD SIGNIF. TEST
0B44    D1CB    CFAD 050                   053   R2GUAR       BR IF S5=1             BR IF GUARD IS R2
0B46    75AD    CFAD 051    R2TRUE                             G1C=G1%T0+C            DO SIG TEST ON R1
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS
0B48  8B4E    CFAD 052                 055   CHKNOR     BR
0B4A  C5C6    CFAD 053    R2GUAR       051   R2TRUE     BR IF S0=0               D  TRUE ADD
0B4C  755B    CFAD 054                                  G1C=0-G1+C               R2 GUARD COMP
0B4E  D052    CFAD 055    CHKNOR       057   NORM       BR IF G05=0
0B50  0008    CFAD 056                                  RST S1    ** UNORMALIZED-GUARD NOT TESTED FOR SIGNIFI.
0B52  B04A    CFAD 057    NORM    CFCO 120   COMPUT     BAL                      GO ADD,S,C R1%R2
0B54  5AE2    CFAD 058                                  RDH T DA,BC              GET R1 ADDRESS
0B56  C5EE    CFAD 059                 071   TCOMPU     BR IF S0=0               BR IF TRUE COMPUTE
0B58  F5DE    CFAD 060                 063   NOCARY     BR IF S3=0               COMPLIMENT -CHECK CARRY
0B5A  0002    CFAD 061                                  RST S3
0B5C  8B6E    CFAD 062                 071   TCOMPU     BR                       ALL OK RESULT IS TRUE FORM
0B5E  1E8D    CFAD 063    NOCARY                        H0=H0<K80                INV SIGN-COMP OP -MUST RECOMP
0B60  E070    CFAD 064                 072   CPFIN      BR IF G06=0              BR IF COMPARE
0B62  0008    CFAD 065                                  RST S1                    RST S1 FOR RES CHECK OF COMP
0B64  2002    CFAD 066                                  SET S3                   COMP GUARD
0B66  755B    CFAD 067                                  G1C=0-G1+C
0B68  D06C    CFAD 068                 070   SAVSIG     BR IF G05=0              BR IF NORMAL. SAVE GUARD SIGNIF.
0B6A  0008    CFAD 069                                  RST S1                   UNORM. RST SIG. TEST BIT
0B6C  9564    CFAD 070    SAVSIG  CFCO 103   RECOMP     BAL
0B6E  E07F    CFAD 071    TCOMPU       080   NOTCP      BR IF G06=1              BR IF ADD OR SUB
0B70  8178    CFAD 072    CPFIN   CCOM 066   LRSTRB     BAL
0B72  D5F9    CFAD 073                 077   COMNZ      BR IF S1=1                SET
0B74  F5F9    CFAD 074                 077   COMNZ      BR IF S3=1               BR IF OVERFLOW ON CMP.--NOT-0
0B76  AA74    CFAD 075            CICY 003   CHECKX     BR     ** RES =0              COND
              CFAD 076  *                                                                CODE
0B78  CE7D    CFAD 077    COMNZ        079   COMPNE     BR IF H00=1              BR IF NEG
0B7A  A824    CFAD 078            CLOG 063   OP95A      BR                       SET CC 2
0B7C  A820    CFAD 079    COMPNE  CLOG 061   OP95C  0   BR                       SET CC 1
0B7E  F5A9    CFAD 080    NOTCP        104   FRACOV     BR IF S3=1               BR IF CARR RES NOT 0
              CFAD 081  *                                                           FRACTION MUST SHIFT RIGHT  4
0B80  5CC2    CFAD 082                                  RDH P DA,B8              RESTORE P
0B82  D5CA    CFAD 083                 123   FRACZO     BR IF S1=0               GO CHECK SIG MASK
0B84  D043    CFAD 084                 118   UNOREN     BR IF G05=1
0B86  0FF3    CFAD 085    CONORM                        Z=H1*-K0F                CHECK HI DIGIT FOR NORMALIZING
0B88  C4C2    CFAD 086                 118   UNOREN     BR IF ZNZ                BR IF NO NORMALIZE REQUIRED
0B8A  B2F8    CFAD 087            CFCO 034   SHIFL4     BAL
0B8C  2EFF    CFAD 088                                  H0=H0+KFF                DECREMENT EXPONENT
0B8E  E496    CFAD 089                 093   NOOVER     BR IF NOVFL
0B90  F499    CFAD 090                 094   UNDERF     BR IF AC=1
0B92  2507    CFAD 091    HA1BC                         G1=0  **  ZERO GUARD-HAVE 1BC NO UNDERFLOW
0B94  8B86    CFAD 092                 085   CONORM     BR
0B96  F493    CFAD 093    NOOVER       091   HA1BC      BR IF AC=1
0B98  ED1F    CFAD 094    UNDERF       099   TAKEIN     BR IF P12=1                    EXPONENT HAS UNDERFLOWED
              CFAD 095  *                                                                IF UNDF IS MASKED RESULT IS 0'D OUT
              CFAD 096  *                                                                IF NOT MASKED CONT NORM. BUT TAKE INTER.
0B9A  2C07    CFAD 097                                  P0=0                     SET CC0
0B9C  B1A0    CFAD 098            CFHA 029   DIVNOI     BR  ** UNDERFLOW BUT NO INT,GO STORE 0,REST I,GO ICYC
0B9E  2080    CFAD 099    TAKEIN                        SET S4                   REMEMBER INT
0BA0  1E8D    CFAD 100                                  H0=H0<K80                INVERT SIGN
0BA2  27D3    CFAD 101                                  D1=0$K0D                 UNDERFLOW INT CODE
0BA4  2507    CFAD 102                                  G1=0                     ZERO OUT GUARD
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0BA6    8B86    CFAD 103                  085     CONORM     BR                 CONTINUE NORMALIZING
0BA8    2A13    CFAD 104    FRACOV                           T0=0$K01           FRAC.HAS OVERF. INSERT 01 FOR
                CFAD 105    *                                                     RIGHT SHIFT
0BAA    2C13    CFAD 106                                     P0=0$K01           SET SHIFT COUNT =1
0BAC    A656    CFAD 107              CFCO 007   SHIFR4      BAL
0BAE    5CC2    CFAD 108                                     RDH P DA,B8        RESTORE P
0BB0    2A07    CFAD 109                                     T0=0
0BB2    2E1B    CFAD 110                                     H0=H0+K01          INC EXPONENT
0BB4    E4BB    CFAD 111                  114     ACBRAN     BR IF OVFL          CHECK
0BB6    F4BD    CFAD 112                  115     EXPOV      BR IF AC=1           FOR EXPONENT OVERFLOW
0BB8    8BC2    CFAD 113    NO1BC         118     UNOREN     BR  ** NO 1BC-NO OVERFLOW-GO SET CC+STORE RESULT
0BBA    F4B9    CFAD 114    ACBRAN        113     NO1BC      BR IF AC=1
0BBC    1E8D    CFAD 115    EXPOV                            H0=H0<K80          INVERT SIGN
0BBE    27C3    CFAD 116                                     D1=0$K0C           SET OVERF INT CODE
0BC0    208C    CFAD 117                                     SET S4             REMEMBER  INT TO BE TAKEN
0BC2    2C25    CFAD 118    UNOREN                           P0=0$K20           SET CC 2  +
0BC4    CE48    CFAD 119                  121     FIN        BR IF H00=0        IF POS CC IS CORRECT
0BC6    1C7D    CFAD 120                                     P0=P0<K70          RES - CORRECT CC = 1
0BC8    B1A6    CFAD 121    FIN       CFHA 032   STORE1      BR                 STORE RESULT,RESTORE I , GO TO
                CFAD 122    *                                                     I CYCLES OR INTER. IF S4=1
0BCA    FD4F    CFAD 123    FRACZO        125     SIGINT     BR IF P13=1        BR IF SIG INT NOT MASKED
0BCC    2E07    CFAD 124                                     H0=0               ZERO OUT EXP -NO INT
0BCE    1F85    CFAD 125    SIGINT                           H0=H0*-K80         MAKE TRUE ZERO
0BD0    896A    CFAD 126              CFCO 078   STORE2      BAL                STORE 0 RES
0BD2    817A    CFAD 127              CCOM 067   LRSTRC      BAL                RESTORE I
0BD4    2C07    CFAD 128                                     P0=0                        C C C
0BD6    FD5B    CFAD 129                  131     TSIGIN     BR IF P13=1
0BD8    AA76    CFAD 130              CICY 004   CHECK      BR
0BDA    2BE3    CFAD 131    TSIGIN                           T1=0$K0E
0BDC    89FC    CFAD 132              BPSW 003   PROG       BR
                CFAD 133    AEND
```

****
* CROSS REFERENCE FOR CSECT CFAD *
****

```
CFAD 007    CFCY 061  CFCY 062  CFCY 063  CFCY 066  CFCY 067
CFAD 020    CFAD 022
CFAD 022    CFAD 018
CFAD 023    CFAD 019
CFAD 027    CFAD 021
CFAD 029    CFAD 027
CFAD 031    CFAD 029
CFAD 034    CFAD 032
CFAD 043    CFAD 035
CFAD 046    CFAD 044
CFAD 047    CFAD 034
CFAD 048    CFAD 042
CFAD 051    CFAD 053
CFAD 053    CFAD 050
CFAD 055    CFAD 052
CFAD 057    CFAD 055
CFAD 063    CFAD 060
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT CFAD *
                                  **********************************

CFAD 070     CFAD 068
CFAD 071     CFAD 059  CFAD 062
CFAD 072     CFAD 064
CFAD 077     CFAD 073  CFAD 074
CFAD 079     CFAD 077
CFAD 080     CFAD 071
CFAD 085     CFAD 092  CFAD 103
CFAD 091     CFAD 093
CFAD 093     CFAD 089
CFAD 094     CFAD 090
CFAD 099     CFAD 094
CFAD 104     CFAD 080
CFAD 113     CFAD 114
CFAD 114     CFAD 111
CFAD 115     CFAD 112
CFAD 118     CFAD 084  CFAD 086  CFAD 113
CFAD 121     CFAD 119
CFAD 123     CFAD 083
CFAD 125     CFAD 123
CFAD 131     CFAD 129
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                CFC0 001    T                                *** COMMON FLOATING POINT ROUTINES ***
                CFC0 002    *
                CFC0 003    *
                CFC0 004    *                                ***  SHIFT RIGHT 4 ***
                CFC0 005    *
                CFC0 006    *
2656    F45C    CFC0 007     SHIFR4         010   RLONG      BR IF G03=0            BR IF LONG
2658    4953    CFC0 008                                     G1=I1XH+G1L
265A    F46F    CFC0 009                    019   RSHORT     BR IF G03=1            UNC BRANCH TO SHORT OP
265C    4153    CFC0 010     RLONG                           G1=U1XH+G1L            SAVE GUARD LONG
265E    5115    CFC0 011                                     U1=U1XL
2660    4013    CFC0 012                                     U1=U0XH+U1L
2662    5005    CFC0 013                                     U0=U0XL
2664    4303    CFC0 014                                     U0=V1XH+U0L
2666    5335    CFC0 015                                     V1=V1XL
2668    4233    CFC0 016                                     V1=V0XH+V1L
266A    5225    CFC0 017                                     V0=V0XL
266C    4923    CFC0 018                                     V0=I1XH+V0L
266E    5995    CFC0 019     RSHORT                          I1=I1XL
2670    4893    CFC0 020                                     I1=I0XH+I1L
2672    5885    CFC0 021                                     I0=I0XL
2674    4F83    CFC0 022                                     I0=H1XH+I0L
2676    5FF5    CFC0 023                                     H1=H1XL
2678    4AF3    CFC0 024                                     H1=T0XH+H1L
267A    2CFF    CFC0 025                                     P0=P0+KFF              DEC SHIFTCOUNT
267C    C4D6    CFC0 026                    007   SHIFR4     BR IF ZNZ
267E    128E    CFC0 027                                     RTN
                CFC0 028    *
                CFC0 029    *
                CFC0 030    *
                CFC0 031    *                                 *** SHIFT LEFT 4  ***
                CFC0 032    *
                CFC0 033    *
32F8    5FF3    CFC0 034     SHIFL4                          H1=H1XH
32FA    48F5    CFC0 035                                     H1=I0XL$H1H
32FC    5883    CFC0 036                                     I0=I0XH
32FE    4985    CFC0 037                                     I0=I1XL$I0H
3300    5993    CFC0 038                                     I1=I1XH
3302    F408    CFC0 039                    042   LONGL      BR IF G03=0            BR IF LONG
3304    4595    CFC0 040                                     I1=G1XL$I1H            SHIFT IN GUARD
3306    128E    CFC0 041                                     RTN
3308    4295    CFC0 042     LONGL                           I1=V0XL$I1H
330A    5223    CFC0 043                                     V0=V0XH
330C    4325    CFC0 044                                     V0=V1XL$V0H
330E    5333    CFC0 045                                     V1=V1XH
3310    4035    CFC0 046                                     V1=U0XL$V1H
3312    5003    CFC0 047                                     U0=U0XH
3314    4105    CFC0 048                                     U0=U1XL$U0H
3316    5113    CFC0 049                                     U1=U1XH
3318    4515    CFC0 050                                     U1=G1XL$U1H            SHIFT IN LONG GUARD
331A    128E    CFC0 051                                     RTN
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS

                CFC0 052   *
                CFC0 053   *
                CFC0 054   *
                CFC0 055   *                              ***  ZERO  TEST  ***
3170   0008     CFC0 056    ZTEST                         RST S1
3172   6FF5     CFC0 057                                  H1=H1$H1                MOVE THRU ALU -SET S1 IF NZ
3174   4886     CFC0 058                                  I=I
3176   F47D     CFC0 059                      062  ZSHORT  BR IF G03=1             BR IF SHORT
3178   4226     CFC0 060                                  V=V
317A   4006     CFC0 061                                  U=U
317C   128E     CFC0 062    ZSHORT                        RTN
                CFC0 063   *
                CFC0 064   *
                CFC0 065   *
                CFC0 066   *
                CFC0 067   *
                CFC0 068   *                              ***   ZERO  OUT  RESULT  ***
166C   4E52     CFC0 069    ZEROUT                        RDH H DC,9A             READ ZERO CONSTANT INTO H
                CFC0 070   *
166E   48E6     CFC0 071                                  I=H
1670   42E6     CFC0 072                                  V=H
1672   40E6     CFC0 073                                  U=H
1674   128E     CFC0 074                                  RTN
                CFC0 075   *
                CFC0 076   *
                CFC0 077   *                              ***   STORE ROUTINE    ***
096A   7EA8     CFC0 078    STORE2                        STH H AS,T+2
096C   78A8     CFC0 079                                  STH I AS,T+2
096E   F475     CFC0 080                      083  SHORTS  BR IF G03=1             BR IF SHORT
0970   72A8     CFC0 081                                  STH V AS,T+2
0972   70A0     CFC0 082                                  STH U AS,T+0
0974   1B73     CFC0 083    SHORTS                        T1=T1*-K07              RST T TO 00X8
0976   128E     CFC0 084                                  RTN
                CFC0 085   *
                CFC0 086   *
                CFC0 087   *
                CFC0 088   *
                CFC0 089   *                              ***  READ  REGISTER  ***
342E   5EA8     CFC0 090    RDOPER                        RDH H AS,T+2
3430   58A8     CFC0 091                                  RDH I AS,T+2
3432   F439     CFC0 092                      095  SHORTR  BR IF G03=1             BR IF SHORT
3434   52A8     CFC0 093                                  RDH V AS,T+2
3436   50A0     CFC0 094                                  RDH U AS,T+0
3438   1B73     CFC0 095    SHORTR                        T1=T1*-K07              T1= X8
343A   128E     CFC0 096                                  RTN
                CFC0 097   *
                CFC0 098   *
                CFC0 099   *
                CFC0 100   *                              ***  COMPLIMENT ROUTINE  ***
                CFC0 101   *
                CFC0 102   *
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
1564   F46F    CFC0 103    RECOMP      108    SHORTC     BR IF G03=1        BR IF SHORT
1566   711B    CFC0 104                                  U1C=0-U1+C
1568   700B    CFC0 105                                  U0C=0-U0+C
156A   733B    CFC0 106                                  V1C=0-V1+C
156C   722B    CFC0 107                                  V0C=0-V0+C
156E   799B    CFC0 108    SHORTC                        I1C=0-I1+C
1570   788B    CFC0 109                                  I0C=0-I0+C
1572   7FFB    CFC0 110                                  H1C=0-H1+C
1574   E079    CFC0 111                113   COMPAR     BR IF G06=1        BR IF NOT COMPARE OP
1576   0002    CFC0 112                                  RST S3
1578   128E    CFC0 113    COMPAR                        RTN
               CFC0 114   *
               CFC0 115   *
               CFC0 116   *
               CFC0 117   *                              ***  ADD AND DIVIDE  COMUTE  R1-R2 ***
               CFC0 118   *
               CFC0 119   *
304A   3BE3    CFC0 120    COMPUT                        T1=T1$K0E          SET T TO LOW HALF 00XE
304C   F452    CFC0 121                124   LOCOMP     BR IF G03=0        BR IF LONG
304E   1B43    CFC0 122                                  T1=T1*-K04         T1 = 00XA  FOR SHORT LOW
3050   F45F    CFC0 123                130   SHCOMP     BR IF G03=1        UNC BRANCH
3052   56AA    CFC0 124    LOCOMP                        RDH D AS,T-2        T= 00XE       T IS SET
3054   717D    CFC0 125                                  U1C=U1%D1+C                        TO 38 OR 78
3056   706D    CFC0 126                                  U0C=U0%D0+C                         WHICH IS A
3058   56AA    CFC0 127                                  RDH D AS,T-2        T= 00XC          SAVE AREA
305A   737D    CFC0 128                                  V1C=V1%D1+C                                FOR R2 OP
305C   726D    CFC0 129                                  V0C=V0%D0+C
305E   56AA    CFC0 130    SHCOMP                        RDH D AS,T-2        T= 00XA
3060   797D    CFC0 131                                  I1C=I1%D1+C
3062   786D    CFC0 132                                  I0C=I0%D0+C
3064   56A0    CFC0 133                                  RDH D AS,T+0        T= 00X8
3066   7F7D    CFC0 134                                  H1C=H1%D1+C
3068   C1ED    CFC0 135                137   DIVCOM     BR IF S4=1         BR IF DIVIDE
306A   128E    CFC0 136                                  RTN                A,S,C
306C   7E6D    CFC0 137    DIVCOM                        H0C=H0%D0+C
306E   128E    CFC0 138                                  RTN
               CFC0 139   *
               CFC0 140   *
               CFC0 141   *                              ***  RX OPERAND  FETCH  ***
               CFC0 142   *
3454   5E18    CFC0 143    RXOPER                        RDH H U+2
3456   5818    CFC0 144                                  RDH I U+2
3458   F45F    CFC0 145                148   XSHORT     BR IF G03=1
345A   5218    CFC0 146                                  RDH V U+2
345C   5010    CFC0 147                                  RDH U U+0
345E   128E    CFC0 148    XSHORT                        RTN
```

* CROSS REFERENCE FOR CSECT CFCO *

```
CFCO 007    CFAD 038  CFAD 046  CFAD 107  CFCO 026  CFDV 016  CFHA 010
CFCO 010    CFCO 007
CFCO 019    CFCO 009
CFCO 034    CFAD 087  CFHA 016  CFMD 030  CFMU 046
CFCO 042    CFCO 039
CFCO 056    CFDV 048  CFLS 025  CFMD 010  CFMU 051
CFCO 062    CFCO 059
CFCO 069    CFHA 030
CFCO 078    CFAD 008  CFAD 039  CFAD 126  CFHA 032  CFMD 037  CFMD 051
CFCO 083    CFCO 080
CFCO 090    CFAD 011  CFAD 037  CFAD 041  CFCY 019  CFDV 009  CFDV 047  CFLS 036  CFMD 039  CFMD 060  CFMD 068
CFCO 095    CFCO 092
CFCO 103    CFAD 070
CFCO 108    CFCO 103
CFCO 113    CFCO 111
CFCO 120    CFAD 057  CFDV 007  CFDV 026
CFCO 124    CFCO 121
CFCO 130    CFCO 123
CFCO 137    CFCO 135
CFCO 143    CFCY 044
CFCO 148    CFCO 145
```

CFCY DESCRIPTIVE TEXT

THE FLOATING POINT I-CYCLES ROUTINE -CFCY- IS ENTERED AT LABELS -RROP- OR -RXOP- FROM THE MAIN I-CYCLES ROUTINE -ICYC-.

THE CHECK FOR CORRECT OPERAND ADDRESS BOUNDRIES AND THE CHECK FOR VALID FLOATING POINT REGISTER ADDRESSES IS MADE.

THE SECOND OPERAND IS FETCHED AND THE FIRST OPERAND ADDRESS IS DEVELOPED BEFORE A BRANCH IS TAKEN TO THE INDIVIDUAL FLOATING POINT OPERATION ROUTINES.

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CFCY 001   T                                  FLOATING POINT  I CYCLE DECODE  RR AND RX
              CFCY 003   *                                                      *
              CFCY 004   *                ALL OPS EXCEPT STORE BEGIN ALIKE  AS FOLLOWS
              CFCY 005   *                2ND OPERAND IS IN LS  H0,H1 I0,I1,G1              (SHORT)
              CFCY 006   *                                      H0,H1,I0,I1,V0,V1,U0,U1,G1  (LONG )
              CFCY 007   *                1ST OPERAND-NORM.OR UNNOR. IS IN R1 LOCATION
              CFCY 008   *                                   *** RR  I CYCLES  FLTPT ***
28C4    E1C9  CFCY 012    RROP          014  NOEXEC    BR IF S6=1          CHK EXECUTE STAT BR IF NOT
28C6    3485  CFCY 013                                  G0=G0$K80           REMEMBER EXECUTE
28C8    0567  CFCY 014    NOEXEC                       Z=G1*-K66           CHEC FOR ODD REG=SPEC CHECK
28CA    C4C2  CFCY 015                051  SPEC      BR IF ZNZ
28CC    9814  CFCY 016          CCOM  057  LSAVEA    BAL                 SAVE P,I,G
28CE    55B3  CFCY 017                               T1=G1XH             R2 ADDRESS
28D0    3883  CFCY 018                               T1=T1$K08
28D2    B42E  CFCY 019          CFCO  090  RDOPER    BAL                 GET 2ND OP- H0H1I0I1  (SHORT)
28D4    45BB  CFCY 020    *                                                     H0H1I0I1V0V1U0U1 (LONG )
              CFCY 021    R1OPER                     T1=G1H+T1L          R1 ADDRESS=T1 X8
28D6    2507  CFCY 022                               G1=0
28D8    7AE2  CFCY 023                               STH T DA,BC         SAVE R1 OPER ADD IN AUX
28DA    10CE  CFCY 024                               RST S K=FC
28DC    B011  CFCY 025                052  FLOAT  N  N=G0L               SEP. OPS
              CFCY 026    *                           **********************
              CFCY 031    *                           *** RX  I CYCLES  FLTPT ***
              CFCY 032    *
              CFCY 033    *
28A0    A6D8  CFCY 034    RXOP      CICY 092  BASE2     BAL                 GET REST OF OP
28A2    2B83  CFCY 035                               T1=0$K08
28A4    9814  CFCY 036          CCOM  057  LSAVEA    BAL                 SAVE P,I,G
28A6    0565  CFCY 037                               Z=G1*-K60           CHK IF R2 NOT 0,2,4,6
28A8    E0C2  CFCY 038                051  SPEC      BR IF HZNZ
28AA    AB25  CFCY 039          CICY  056  ADDERR N   BR IF H1=NZ         CHK 2ND OPER  ADD  TOO LARGE
28AC    F43F  CFCY 040                049  SHORT     BR IF G03=1         BR IF SHORT OP
28AE    0183  CFCY 041                               Z=U1*-K08           CHK FOR DOUBLE BOUNDARY
28B0    F0C2  CFCY 042                051  SPEC      BR IF LZNZ
28B2    C038  CFCY 043    LOOKLD                051  STORCK    BR IF G04=0
28B4    B454  CFCY 044          CFCO  143  RXOPER    BAL
28B6    A8D4  CFCY 045                021  R1OPER    BR
28B8    04F5  CFCY 046    STORCK                     Z=G0*-KF0           LOOK FOR 60 OR 70 OP
28BA    F08A  CFCY 047                057  FLOAT  5  BR IF LZNZ          OPER NOT STORE OP
28BC    8924  CFCY 048          CFLS  034  STORE     BR
28BE    01C3  CFCY 049    SHORT                      Z=U1*-K0C           CHK WORD BOUNDARY
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

28C0    F0B3    CFCY 050                  043   LOOKLD     BR IF LZ=0
28C2    AE94    CFCY 051    SPEC      CICY 070  SPECHK     BR
2880    A150    CFCY 052    FLOAT  0  CFLS 007  LOADP      BR                 LOAD POS
2882    9BEE    CFCY 053    FLOAT  1  CFLS 014  LOADN      BR                 LOAD NEG
2884    9BF2    CFCY 054    FLOAT  2  CFLS 022  LOADT      BR                 LOAD AND TEST
2886    9BF0    CFCY 055    FLOAT  3  CFLS 021  LOADC      BR                 LOAD COMPLIMENT
2888    B17E    CFCY 056    FLOAT  4  CFHA 006  HALVE      BR                 HALVE
288A    B484    CFCY 057    FLOAT  5       068  NOGOOD     BR                 OP ERROR
288C    B484    CFCY 058    FLOAT  6       068  NOGOOD     BR                 OP ERROR
288E    B484    CFCY 059    FLOAT  7       068  NOGOOD     BR                 OP ERROR
2890    B1A6    CFCY 060    FLOAT  8  CFHA 032  STORE1     BR                 LOAD
2892    8AF2    CFCY 061    FLOAT  9  CFAD 007  SCAD       BR                 COMPARE-LONG,SHORT-RR OR  RX
2894    8AF2    CFCY 062    FLOAT  A  CFAD 007  SCAD       BR                 ADD NORMALIZED LONG OR  SHORT
2896    8AF2    CFCY 063    FLOAT  B  CFAD 007  SCAD       BR                 SUBTRACT NORM  LONG OR  SHORT
2898    9E00    CFCY 064    FLOAT  C  CFMD 010  MULPRE     BR                 MULTIPLY   LONG OR SHORT
289A    9DFE    CFCY 065    FLOAT  D  CFMD 009  DIVPRE     BR                 DIVIDE     LONG OR SHORT
289C    8AF2    CFCY 066    FLOAT  E  CFAD 007  SCAD       BR                 ADD UNORMALIZED LONG OR SHORT
289E    8AF2    CFCY 067    FLOAT  F  CFAD 007  SCAD       BR                 SUBTRACT UNORMALIZED LONG OR SH
3484    817A    CFCY 068    NOGOOD    CCOM 067  LRSTRC     BAL                RESTORE I
3486    83B4    CFCY 069              CICY 077  OPERR      BR
                                      ************************************
                                      * CROSS REFERENCE FOR CSECT CFCY *
                                      ************************************

CFCY 012    CICY 015  CICY 016
CFCY 014    CFCY 012
CFCY 021    CFCY 045
CFCY 034    CICY 019  CICY 020
CFCY 043    CFCY 050
CFCY 046    CFCY 043
CFCY 049    CFCY 040
CFCY 051    CFCY 015  CFCY 038  CFCY 042
CFCY 052    CFCY 025
CFCY 057    CFCY 047
CFCY 068    CFCY 057  CFCY 058  CFCY 059
```

CFDV DESCRIPTIVE TEXT

THE FLOATING POINT DIVIDE INSTRUCTIONS ARE FIRST HANDLED BY THE -CFMD- ROUTINE. THERE THE ZERO FRACTION TEST, PRENORMALIZATION, AND SUBTRACTION OF EXPONENTS IS MADE.

THE -CFDV- ROUTINE HANDLES THE COMPUTATION OF THE QUOTENT AND THE STORING OF THE RESULTS. THE -CFCO- ROUTINE IS USED FOR ACTUAL COMPUTATION, RIGHT SHIFTING, AND ZERO TESTING OF THE RESULT. THE -CFHA- ROUTINE HANDLES LEFT SHIFTING, INTERRUPT CHECKING, AND THE DIVIDE EXIT TO EITHER THE INTERRUPT OR I-CYCLES ROUTINE.

OP CODES HANDLED BY THE -CFDV- ROUTINE

OP CODE TYPE	BYTE 1	BYTE 2	BYTE 3	BYTE 4
RR DIVIDE (LONG)	2D	R1 R2		
RR DIVIDE (SHORT)	3D	R1 R2		
RX DIVIDE (LONG)	6D	R1 X2	B2 D2	D2
RX DIVIDE (SHORT)	7D	R1 X2	B2 D2	D2

OBJECTIVE -- THE FIRST OPERAND IS DIVIDED BY THE SECOND OPERAND. THE QUOTIENT REPLACES THE FIRST OPERAND.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

              CFDV 001    T                                  FLOATING POINT DIVIDE
              CFDV 002    *
              CFDV 003    *
              CFDV 004    *
              CFDV 005    *
1F82    3082  CFDV 006    DIVIDE                             SET S K=98           SET S0,S3,S4
1F84    B04A  CFDV 007                       CFCO 120   COMPUT  BAL              DO TRIAL SUBTRACT
1F86    5AE2  CFDV 008                                          RDH T DA,BC
1F88    B42E  CFDV 009                       CFCO 090   RDOPER  BAL              GET R1
1F8A    46A6  CFDV 010                                          D=T
1F8C    3B35  CFDV 011                                          T1=T1$K30        T= R2 SAVE 0038 OR 0078
1F8E    3713  CFDV 012                                          D1=D1$K01        D=QUOTIENT ADD 00R19
1F90    76F2  CFDV 013                                          STH D DA,BE      SAVE QUOT ADD
1F92    F5A2  CFDV 014                            022   NCNOSH  BR IF S3=0
1F94    2C13  CFDV 015                                          P0=0$K01         R1 MUST SHIFT RIGHT ONE DIGIT
1F96    A656  CFDV 016                       CFCO 007   SHIFR4  BAL
1F98    5CA2  CFDV 017                                          RDH P DA,AC      GET RESULT EXPONENT
1F9A    2C1B  CFDV 018                                          P0=P0+K01        INC EXP
1F9C    C4A0  CFDV 019                            021   FIXUND  BR IF ZNZ        CHK FOR PREV.UNDERFLOW
1F9E    1D13  CFDV 020                                          P1=P1*-K01        UNDERFLOW-CORRECTED RST IND.
1FA0    7CA2  CFDV 021    FIXUND                               STH P DA,AC      PUT ADJ. RES. EXPONENT BACK
1FA2    9DE0  CFDV 022    NCNOSH  CFHA 044   DIVIDL  BAL                       SHIFT LEFT 1
1FA4    3002  CFDV 023                                          SET S K=90       SET S0,S3
1FA6    0040  CFDV 024    LOOP1                                RST S5
1FA8    2D13  CFDV 025                                          P1=0$K01         SET QUOTIENT FLAG
1FAA    B04A  CFDV 026    LOOP    CFCO 120   COMPUT  BAL                       SUB  VISOR  FROM DEND
1FAC    F5EE  CFDV 027                            034   DOADD   BR IF S3=0      IF NO CARRY - DO AN ADD
1FAE    9DE0  CFDV 028                       CFHA 044   DIVIDL  BAL              SHIFT DEND AND QUOT FLAG LEFT 1
1FB0    3002  CFDV 029                                          SET S K=90
1FB2    D1BB  CFDV 030    QUOTCH                   037   STQUOT  BR IF S5=1     BR IF QUOTIENT BYTE FULL
1FB4    CD2A  CFDV 031                            026   LOOP    BR IF P10=0      CONTINUE COMPUTE IF NO QUOT FLAG
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

1FB6    2040    CFDV 032                                   SET S5             7TH TIME-DO 1 MORE THEN STORE QU
1FB8    9FAA    CFDV 033                 026     LOOP      BR
1FEE    9DE0    CFDV 034    DOADD        CFHA 044  DIVIDL  BAL                SHIFT LEFT
1FF0    1002    CFDV 035                                   RST S K=90           ADD  VISOR
1FF2    9FB2    CFDV 036                 030     QUOTCH    BR                     TO DEND
1FBA    56F2    CFDV 037    STQUOT                         RDH D DA,BE        GET QUOT ADDRESS
1FBC    7D68    CFDV 038                                   STB P1 AS,D+1      STORE QUOT BYTE
1FBE    76F2    CFDV 039                                   STH D DA,BE        STORE EQUOT ADDRESS
1FC0    F448    CFDV 040                 044     LONGD     BR IF G03=0
1FC2    07CB    CFDV 041                                   Z=D1<K0C           CHECK FOR 00XC
1FC4    F0A6    CFDV 042                 024     LOOP1     BR IF LZNZ          END OF SHORT
1FC6    9FCC    CFDV 043                 046     ENDDIV    BR                 SHORT DIV FINISHED
1FC8    5779    CFDV 044    LONGD                          D1=D1
1FCA    F0A6    CFDV 045                 024     LOOP1     BR IF LZNZ         BR IF LONG NOT FINISHED
1FCC    5AE2    CFDV 046    ENDDIV                         RDH T DA,BC        GE R1 ADDRESS
1FCE    B42E    CFDV 047             CFCO 090  RDOPER     BAL                GET QUOTIENT
1FD0    B170    CFDV 048             CFCO 056  ZTEST      BAL
1FD2    D5D7    CFDV 049                 051     GETEXP    BR IF S1=1         BR IF QUOT NOT ZERO
1FD4    B1A0    CFDV 050             CFHA 029  DIVNOI     BR                 RESULT IS 0 STORE 0,GET I,GO,ICY
1FD6    56A2    CFDV 051    GETEXP                         RDH D DA,AC        DO=EXP-----D1=SIGN+FLAGS
1FD8    5659    CFDV 052                                   G1=D0              MOVE RES EXP FOR CORREC+STORE
1FDA    5CC2    CFDV 053                                   RDH P DA,B8        RESTORE MASK
1FDC    1585    CFDV 054                                   G1=G1*-K80         MAKE RES SIGN +
1FDE    C762    CFDV 055                 057     STOEXP    BR IF D10=0        BR IF RES IS  +
1FE0    3585    CFDV 056                                   G1=G1$K80          SIGN - MAKE RESULT -
1FE2    75A0    CFDV 057    STOEXP                         STB G1 AS,T+0
1FE4    F375    CFDV 058                 063     UNDERF    BR IF D17=1          BR IF UNDERFLOW FLAG IS ON
1FE6    C66B    CFDV 059                 061     OVFLOW    BR IF D00=1          BR IF OVERFLOW
1FE8    C080    CFDV 060                                   RST S4               REST INT FLAG IF NO OVERFLOW
1FEA    27C3    CFDV 061    OVFLOW                         D1=0$K0C
1FEC    B1A8    CFDV 062             CFHA 033  DIVOVF     BR                   GO RESTORE I ,TAKE INT OR GO ICY
1FF4    27D3    CFDV 063    UNDERF                         D1=0$K0D
1FF6    55E9    CFDV 064                                   H0=G1                MOVE RES EXP FOR UND STORE
1FF8    B19E    CFDV 065             CFHA 028  UNDCHK     BR                   GO CHECK UNDERFLOW MASK
                                     ************************************
                                     * CROSS REFERENCE FOR CSECT CFDV *
                                     ************************************

CFDV 006    CFMD 054
CFDV 021    CFDV 019
CFDV 022    CFDV 014
CFDV 024    CFDV 042  CFDV 045
CFDV 026    CFDV 031  CFDV 033
CFDV 030    CFDV 036
CFDV 034    CFDV 027
CFDV 037    CFDV 030
CFDV 044    CFDV 040
CFDV 046    CFDV 043
CFDV 051    CFDV 049
CFDV 057    CFDV 055
CFDV 061    CFDV 059
CFDV 063    CFDV 058
```

CFHA DESCRIPTIVE TEXT

THE -CFHA- ROUTINE HANDLES THE FOLLOWING OP CODES-

RROP 34 (SHORT OPERANDS)
RXOP 24 (LONG OPERANDS)

THE -CFHA- ROUTINE PERFORMS THE FLOATING POINT HALVE OPERATION. THIS IS DONE BY SHIFTING THE SECOND OPERAND FRACTION LEFT 3 BITS THEN RIGHT 4 BITS. THE RESULT IS NORMALIZED AND STORED IN THE FIRST OPERAND LOCATION.

THE -CFHA- ROUTINE PERFORMS THE EXIT TO I-CYCLES OR THE INTERRUPT ROUTINE FOR THE FLOATING POINT LOAD OPS, AND CERTAIN CASES FOR FLOATING DIVIDE, MULTIPLY, AND ADD OPS.

THE -CFHA- ROUTINE ALSO PERFORMS THE LEFT SHIFT OF THE QUOTIENT FOR THE FLOATING DIVIDE OPERATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFHA 001	T			*** HALVE OP *** FLOATING POINT	
		CFHA 002	*				
		CFHA 003	*				
		CFHA 004	*				
		CFHA 005	*				
317E	2C13	CFHA 006	HALVE			P0=0$K01	SET RIGHT SHIFT COUNT
3180	9DE4	CFHA 007		046	LEFT1	BAL	SHIFT
3182	9DE4	CFHA 008		046	LEFT1	BAL	LEFT
3184	9DE4	CFHA 009		046	LEFT1	BAL	ONE THREE TIMES
3186	A656	CFHA 010		CFCO 007	SHIFR4	BAL	RIGHT 1 = LEFT 3 THEN RIGHT 4
3188	5CC2	CFHA 011				RDH P DA,B8	
318A	D5A4	CFHA 012		031	FRAC00	BR IF S1=0	BR IF 2ND OPER WAS 0
318C	2A07	CFHA 013				T0=0	
318E	0FF3	CFHA 014	HNORM			Z=H1*-K0F	CHECK HI DIGIT
3190	C4A6	CFHA 015		032	STORE1	BR IF ZNZ	
3192	B2F8	CFHA 016		CFCO 034	SHIFL4	BAL	GO NORMALIZE
3194	2507	CFHA 017				G1=0	ZERO OUT GUARD
3196	2FFF	CFHA 018				H0=H0+KFF	DEC EXPONENT
3198	E4B2	CFHA 019		022	NOVFLH	BR IF NOVFL	CHECK
319A	F4B5	CFHA 020		023	HAUND	BR IF AC=1	1BC
319C	B18E	CFHA 021	OK1BC	014	HNORM	BR	FOR
31B2	F49D	CFHA 022	NOVFLH	021	OK1BC	BR IF AC=1	UNDERFLOW NO 1BC=UNDERF
31B4	ED22	CFHA 023	HAUND	030	HZER	BR IF P12=0	BR IF NO INT. TO BE TAKEN
31B6	1E8D	CFHA 024				H0=H0<K80	INVERT THE SIGN BIT
31B8	27D3	CFHA 025				D1=0$K0D	INT CODE SAVE
31BA	2080	CFHA 026				SET S4	REMEMBER INT
31BC	B18E	CFHA 027		014	HNORM	BR	
31BE	ED27	CFHA 028	UNDCHK	032	STORE1	BR IF P12=1	BR IF INT TO BE TAKEN
31A0	0080	CFHA 029	DIVNOI			RST S4	
31A2	966C	CFHA 030	HZER	CFCO 069	ZEROUT	BAL	
31A4	2E07	CFHA 031	FRAC00			H0=0	ZERO OUT EXPONENT
31A6	896A	CFHA 032	STORE1	CFCO 078	STORE2	BAL	
31A8	817A	CFHA 033	DIVOVF	CCOM 067	LRSTRC	BAL	
31AA	C1AF	CFHA 034		036	HTAKE	BR IF S4=1	
31AC	AA76	CFHA 035		CICY 004	CHECK	BR	
31AE	5789	CFHA 036	HTAKE			T1=D1	
31B0	89FC	CFHA 037		BPSW 003	PROG	BR	
		CFHA 038	*				
		CFHA 039	*				

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                CFHA 040    *                                    **  LEFT ONE  FOR DIVIDE AND HALVE **
                CFHA 041    *
                CFHA 042    *
                CFHA 043    *
1DE0    6DDD    CFHA 044     DIVIDL                             P1C=P1+P1+C          SHIFT QUOTIENT FOR DIVIDE
1DE2    655B    CFHA 045                                        G1C=G1+G1            SHIFT DIVIDE GUARD
1DE4    F46F    CFHA 046     LEFT1          051   HSHORT       BR IF G03=1          BR IF SHORT
1DE6    611D    CFHA 047                                        U1C=U1+U1+C
1DE8    600D    CFHA 048                                        U0C=U0+U0+C
1DEA    633D    CFHA 049                                        V1C=V1+V1+C
1DEC    622D    CFHA 050                                        V0C=V0+V0+C
1DEE    699D    CFHA 051     HSHORT                             I1C=I1+I1+C
1DF0    688D    CFHA 052                                        I0C=I0+I0+C
1DF2    6FFD    CFHA 053                                        H1C=H1+H1+C
1DF4    C1FB    CFHA 054                   057   DIV          BR IF S4=1
1DF6    6AAD    CFHA 055                                        T0C=T0+T0+C          HALVE
1DF8    128E    CFHA 056                                        RTN                   ONLY
1DFA    6EED    CFHA 057     DIV                                H0C=H0+H0+C          DIVIDE
1DFC    128E    CFHA 058                                        RTN                     ONLY
                                                    **********************************
                                                    * CROSS REFERENCE FOR CSECT CFHA *
                                                    **********************************

CFHA 006     CFCY 056
CFHA 014     CFHA 021  CFHA 027
CFHA 021     CFHA 022
CFHA 022     CFHA 019
CFHA 023     CFHA 020
CFHA 028     CFDV 065  CFMU 061
CFHA 029     CFAD 098  CFDV 050  CFMD 020  CFMU 053
CFHA 030     CFHA 023
CFHA 031     CFHA 012
CFHA 032     CFAD 121  CFCY 060  CFHA 015  CFHA 028  CFLS 028  CFMU 065
CFHA 033     CFDV 062  CFMD 015
CFHA 036     CFHA 034
CFHA 044     CFDV 022  CFDV 028  CFDV 034
CFHA 046     CFHA 007  CFHA 008  CFHA 009
CFHA 051     CFHA 046
CFHA 057     CFHA 054
```

CFLS DESCRIPTIVE TEXT

THE -CFLS- ROUTINE HANDLES THE FOLLOWING OP CODES-

LOAD POSITIVE -20-,-30-,(SHORT) AND (LONG)
LOAD NEGATIVE -21-,-31-,(SHORT) AND (LONG)
LOAD AND TEST -22-,-32-,(SHORT) AND (LONG)
LOAD COMPLEMENT -23-,-33-,(SHORT) AND (LONG)
STORE -60-,-70-, (SHORT) AND (LONG)

OBJECTIVES-

LOAD POSITIVE — THE SECOND OPERAND IS PLACED IN THE FIRST OPERAND LOCATION WITH THE SIGN FORCED POSITIVE.

LOAD NEGATIVE — THE SECOND OPERAND IS PLACED IN THE FIRST OPERAND LOCATION WITH THE SIGN FORCED NEGATIVE.

LOAD COMPLEMENT — THE SECOND OPERAND IS PLACED IN THE FIRST OPERAND LOCATION WITH THE SIGN INVERTED.

LOAD AND TEST — THE SECOND OPERAND IS PLACED IN THE FIRST OPERAND LOCATION AND THE CONDITION CODE IS SET TO INDICATE THE RESULT STATUS.
CC=0 RESULT FRACTION IS ZERO
CC=1 RESULT IS LESS THAN ZERO
CC=2 RESULT IS GREATER THAN ZERO

STORE — THE FIRST OPERAND IS STORED AT THE SECOND OPERAND LOCATION.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFLS 001	T			LOAD AND STORE OPS FLOATING POINT	
		CFLS 002	*				
		CFLS 003	*				
		CFLS 004	*			*** LOAD POSITIVE **	
		CFLS 005	*				
		CFLS 006	*				
2150	1E85	CFLS 007	LOADP			H0=H0*-K80	MAKE SIGN +
2152	98F2	CFLS 008			022 LOADT	BR	
		CFLS 009	*				
		CFLS 010	*				
		CFLS 011	*			*** LOAD NEGATIVE ***	
		CFLS 012	*				
		CFLS 013	*				
1BEE	1E85	CFLS 014	LOADN			H0=H0*-K80	MAKE THE SIGN PLUS
		CFLS 015	*				FALL THROUGH TO COMPLIMENT
		CFLS 016	*				
		CFLS 017	*				
		CFLS 018	*			*** LOAD COMPLIMENT	
		CFLS 019	*				
		CFLS 020	*				
1BF0	1E8D	CFLS 021	LOADC			H0=H0<K80	INVERT SIGN
1BF2	2C25	CFLS 022	LOADT			P0=0$K20	SET +
1BF4	CE78	CFLS 023			025 OKSOFA	BR IF H00=0	
1BF6	1C7D	CFLS 024				P0=P0<K70	NOT + CHANGE CC TO - 0101 CC=1
1BF8	B170	CFLS 025	OKSOFA	CFCO 056	ZTEST	BAL	GO TEST FRACTION FOR 0
1BFA	D5FF	CFLS 026			028 CODCOR	BR IF S1=1	BR IF FRAC NOT 0 CC IS SET
1BFC	2C07	CFLS 027				P0=0	SET CC=0 FRAC=0
1BFE	B1A6	CFLS 028	CODCOR	CFHA 032	STORE1	BR	GO LOAD R2 TO R1,REST I,GO I CYC
		CFLS 029	*				
		CFLS 030	*				
		CFLS 031	*			*** STORE OP ***	
		CFLS 032	*				

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT              COMMENTS

                CFLS 033    *
0924     4606   CFLS 034     STORE                             D=U                    MOVE DESTINATION ADDRESS
0926     45BB   CFLS 035                                       T1=G1H+T1L             R1 OPERAND ADDR
0928     B42E   CFLS 036                CFCO 090   RDOPER     BAL                    GET R1 OPER
092A     7E78   CFLS 037                                       STH H D+2
092C     7878   CFLS 038                                       STH I D+2
092E     F435   CFLS 039                     042   ENDSTO     BR IF G03=1            BA IF SHORT
0930     7278   CFLS 040                                       STH V D+2
0932     7070   CFLS 041                                       STH U D+0
0934     817A   CFLS 042     ENDSTO     CCOM 067   LRSTRC     BAL                    RESTORE I
0936     AA76   CFLS 043                CICY 004   CHECK      BR
                                                   **********************************
                                                   * CROSS REFERENCE FOR CSECT CFLS *
                                                   **********************************

CFLS 007     CFCY 052
CFLS 014     CFCY 053
CFLS 021     CFCY 055
CFLS 022     CFCY 054  CFLS 008
CFLS 025     CFLS 023
CFLS 028     CFLS 026
CFLS 034     CFCY 048
CFLS 042     CFLS 039
```

CFMD DESCRIPTIVE TEXT

THE -CFMD- ROUTINE IS ENTERED FROM THE FLOATING POINT I-CYCLES ROUTINE -CFCY- TO PREPARE FOR THE EXECUTION OF THE FOLLOWING OP CODES -

DIVIDE		MULTIPL	
RROP	3D	RROP	3C
RROP	2D	RROP	2C
RXOP	7D	RXOP	7C
RXOP	6D	RXOP	6C

PRIOR TO BRANCHING TO THE PROPER EXECUTE ROUTINE, THE -CFMD- ROUTINE DOES THE FOLLOWING--

1. TEST MULTIPLIER OR DIVISOR FOR ZERO.
2. PRENORMALIZE OPERANDS
3. COMPUTE INTERMEDIATE EXPONENT AND SIGN OF THE RESULT.
4. PREPARE OPERANDS, (FOR FP MULTIPLY) , TO SHARE THE BINARY MULTIPLY COMPUTE LOOP IN ROUTINE -CMLT-.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CFMD 001	T			MULTIPLY DIVIDE PRENORMALIZE -FLOATING POINT	
		CFMD 002	*				
		CFMD 003	*				
		CFMD 004	*			2ND OP IS PRENOR. FIRST AND PUT IN A SAVE AREA	
		CFMD 005	*			1ST OP IS THEN PRENOR. AND PUT BACK IN R1 AND	
		CFMD 006	*			RETAINED IN LS H,I,V,U	
		CFMD 007	*				
		CFMD 008	*				
1DFE	3082	CFMD 009	DIVPRE			SET S K=98	SET S0,S3,S4 FOR DIVIDE
1E00	B170	CFMD 010	MULPRE	CFCO 056	ZTEST	BAL	CHECK FRACTION FOR 0
1E02	D595	CFMD 011		021	FNOTZ	BR IF S1=1	BR IF PLIER OR DIVISOR IS NOT 0
1E04	D18D	CFMD 012		016	R1ISO	BR IF S5=1	BR IF R1= 0
1E06	F00C	CFMD 013		016	R1ISO	BR IF G07=0	R2 IS 0 BR IF MULT
1E08	27F3	CFMD 014				D1=0$K0F	IT IS DIVIDE EXCEPTION
1E0A	B1A8	CFMD 015		CFHA 033	DIVOVF	BR	GO TAKE INT AND STOR 0 ,REST I
1E0C	F011	CFMD 016	R1ISO	019	DONE	BR IF G07=1	BR IF DIVIDE
1E0E	1415	CFMD 017				G0=G0*-K10	MULT TURN OFF SHORT
		CFMD 018	*				FULL 0 STORE FOR MULT.
1E10	5CC2	CFMD 019	DONE			RDH P DA,B8	RESTORE P
1E12	B1A0	CFMD 020		CFHA 029	DIVNOI	BR	
1E14	D19C	CFMD 021	FNOTZ	025	R2	BR IF S5=0	BR IF USING R2
1E16	5E69	CFMD 022				D0=H0	USING R1 SAVE ORIG EXP
1E18	3E85	CFMD 023				H0=H0$K80	MAKE EXCESS 192
1E1A	9E22	CFMD 024		028	NORM	BR	
1E1C	5E79	CFMD 025	R2			D1=H0	SAVE R2 ORIG EXP
1E1E	1E85	CFMD 026				H0=H0*-K80	REMOVE SIGN
1E20	2E4D	CFMD 027				H0=H0+K40	MAKE EXCESS 128
1E22	0FF3	CFMD 028	NORM			Z=H1*-K0F	CHECK HI DIGIT FOR 0
1E24	C4AC	CFMD 029		033	FNORM	BR IF ZNZ	BR IF NORMALIZED
1E26	B2F8	CFMD 030		CFCO 034	SHIFL4	BAL	
1E28	2EFF	CFMD 031				H0=H0+KFF	DEC EXP
1E2A	9E22	CFMD 032		028	NORM	BR	
1E2C	D1BF	CFMD 033	FNORM	042	R1D	BR IF S5=1	
1E2E	3B35	CFMD 034				T1=T1$K30	DOING R2 SET T TO SAVE AREA ADD
1E30	5ED9	CFMD 035				P1=H0	SAVE NORM R2 EXP
1E32	2E07	CFMD 036				H0=0	
1E34	896A	CFMD 037		CFCO 078	STORE2	BAL	SAVE NORM R2
1E36	5AE2	CFMD 038				RDH T DA,BC	T=00X8
1E38	B42E	CFMD 039		CFCO 090	RDOPER	BAL	GET R1 OP

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
1E3A    2040    CFMD 040                                   SET S5
1E3C    9E00    CFMD 041                 010    MULPRE     BR
1E3E    5EC9    CFMD 042    R1D                            P0=H0              DOING R1,SAVE NORM R1 EXP
1E40    2E07    CFMD 043                                   H0=0               ZERO OUT OPERAND EXPO
1E42    6761    CFMD 044                                   D1=D1<D0           COMPUTE RES SIGN D00 BIT
1E44    17F3    CFMD 045                                   D1=D1*-K0F         ZERO OUT D1 LOW FOR FLAGS
1E46    7CDD    CFMD 046                                   P0C=P0%P1+C        COMP IF DIV,TRUE IF MULT
1E48    57D9    CFMD 047                                   P1=D1              SAVE RES SIGN
1E4A    F5CF    CFMD 048                 050    NOUND      BR IF S3=1
1E4C    3D13    CFMD 049                                   P1=P1$K01          SET UNDERFLOW FLAG
1E4E    7CA2    CFMD 050    NOUND                          STH P DA,AC        SAVE RES EXP,SIGN AND FLAGS
1E50    896A    CFMD 051             CFCO 078  STORE2     BAL                PUT NORMALIZED R1 BACK IN R1
1E52    3B35    CFMD 052                                   T1=T1$K30          T=R2 SAVE  0038 OR 0078
1E54    F058    CFMD 053                 055    NDIV       BR IF G07=0        BR IF MULT
1E56    9F82    CFMD 054             CFDV 006  DIVIDE     BR
1E58    10CE    CFMD 055    NDIV                           RST S K=FC         CONTINUE MULT PREP.
1E5A    2080    CFMD 056                                   SET S4             FOR BINARY MULT
1E5C    F46E    CFMD 057                 066    LONGM      BR IF G03=0
1E5E    42E6    CFMD 058                                   V=H                SAVE
1E60    4086    CFMD 059                                   U=I                 SHORT CAND R1
1E62    B42E    CFMD 060             CFCO 090  RDOPER     BAL                GET PLIER R2
1E64    2B07    CFMD 061                                   T1=0               SHIFT
1E66    46E6    CFMD 062                                   D=H                 DATA
1E68    4E86    CFMD 063                                   H=I                  TO SHARE
1E6A    48A6    CFMD 064                                   I=T                   BINARY
1E6C    8992    CFMD 065             CMLT 027  FLTMU1     BR                 BR TO SHARE BINARY MULT
1E6E    4C26    CFMD 066    LONGM                          P=V                SAVE
1E70    4606    CFMD 067                                   D=U                 LOW CAND LONG  R1
1E72    B42E    CFMD 068             CFCO 090  RDOPER     BAL                GET LONG PLIER R2
1E74    4A86    CFMD 069                                   T=I                SHIFT
1E76    48E6    CFMD 070                                   I=H                 DATA
1E78    4E06    CFMD 071                                   H=U                  TO
1E7A    4066    CFMD 072                                   U=D                   SHARE
1E7C    4626    CFMD 073                                   D=V                    BINARY
1E7E    42C6    CFMD 074                                   V=P                     MULT
1E80    8992    CFMD 075             CMLT 027  FLTMU1     BR                 BR TO SHARE BINARY MULT
                                     ************************************
                                     * CROSS REFERENCE FOR CSECT CFMD *
                                     ************************************
```

```
CFMD 009    CFCY 065
CFMD 010    CFCY 064  CFMD 041
CFMD 016    CFMD 012  CFMD 013
CFMD 019    CFMD 016
CFMD 021    CFMD 011
CFMD 025    CFMD 021
CFMD 028    CFMD 024  CFMD 032
CFMD 033    CFMD 029
CFMD 042    CFMD 033
CFMD 050    CFMD 048
CFMD 055    CFMD 053
CFMD 066    CFMD 057
```

CFMU DESCRIPTIVE TEXT

FLOATING MULTIPLY OP CODES

RROP	3C
RROP	2C
RXOP	7C
RXOP	6C

THE FLOATING POINT MULTIPLY ROUTINE, -CFMU- IS ENTERED FROM THE BINARY MULTIPLY ROUTINE -CMLT-. THE FOLLOWING SEQUENCE HAS TAKEN PLACE PRIOR TO ENTRY TO THE -CMPU- ROUTINE--

- CFCY- THE MULTIPLY OP CODE IS DECODED AND A BRANCH TO THE -CFMD- ROUTINE IS TAKEN.
- CFMD- THE OPERANDS ARE PRENORMALIZED AND SAVED. THE INTERMEDIATE PRODUCT EXPONENT IS COMPUTED AND SAVED. BRANCH TO THE -CLMT- ROUTINE.
- CLMT- COMPUTE INTERMEDIATE PRODUCT. BRANCH TO THE -CFMU- ROUTINE.
- -CFMU- POSTNORMALIZE THE PRODUCT IF NECESSARY. CHECK FOR UNDERFLOW-
  - IF UNDERFLOW, SET UNDERFLOW FLAG AND BRANCH TO THE -CFHA- ROUTINE TO SEE IF AN INTERRUPT IS TO BE TAKEN. IF INTERRUPT, ZERO OUT RESULT FRACTION AND EXPONENT AND GO TO THE -BPSW- ROUTINE TO STORE INTERRUPT CODE IN PSW. IF NO INTERRUPT, GO TO THE NORMAL ENTRY TO THE -CFHA- ROUTINE.
  - IF NO UNDERFLOW, GO TO THE -CFHA- ROUTINE TO STORE THE NORMALIZED PRODUCT IN THE FIRST OPERAND LOCATION. RETURN TO I-CYCLES ROUTINE -CICY-.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

          CFMU 001    T                                      FLOATING POINT MULTIPLY
          CFMU 002    *
          CFMU 003    *
          CFMU 004    *                                      THIS ROUTINE IS ONLY REACHED BY A BRANCH FROM
          CFMU 005    *                                      BINARY MULTIPLY -IF SHORT MULT IS FINISHED,IF LONG
          CFMU 006    *                                       THIS ROUTINE GOES BACK TO BINARY FOR SECOND HALF
          CFMU 007    *
          CFMU 008    *
2354    5CB2  CFMU 009     MULT                              RDH P DA,AE          RESTORE OP CODE TO P
2356    FC08  CFMU 010                          024  LONG     BR IF P03=0
2358    44C6  CFMU 011                                       G=P  ** THIS IS SHORT MULT IS FIN. PUT OP CODE IN G
235A    1415  CFMU 012                                       G0=G0*-K10           MAKE PRODUCT LONG
235C    5CA2  CFMU 013                                       RDH P DA,AC          GET RESULT EXPONENT AND FLAGS
235E    5079  CFMU 014                                       D1=U0
2360    5F09  CFMU 015                                       U0=H1                 SHIFT
2362    57F9  CFMU 016                                       H1=D1                  DATA
2364    5189  CFMU 017                                       I0=U1                   FOR
2366    5299  CFMU 018                                       I1=V0                    POSTNORMALIZE
2368    5329  CFMU 019                                       V0=V1
236A    5E39  CFMU 020                                       V1=H0
236C    2107  CFMU 021                                       U1=0
236E    2507  CFMU 022                                       G1=0                 ZERO OUT GUARD
2370    A32C  CFMU 023                          042  MULTEN   BR
2308    5CA2  CFMU 024     LONG                              RDH P DA,AC          GET RESULT EXPONENT AND FLAGS
230A    C918  CFMU 025                          033  LONGEN   BR IF P14=1          BR IF SECOND TIME
230C    3D83  CFMU 026                                       P1=P1$K08            TURN ON SECOND FLAG
230E    7CA2  CFMU 027                                       STH P DA,AC          PUT FLAGS AND EXPONENT BACK
2310    5CE2  CFMU 028                                       RDH P DA,BC          GET R1 ADD
2312    52C8  CFMU 029                                       RDH V AS,P+2         GET CAND HI  R1
2314    5EC0  CFMU 030                                       RDH H AS,P+0
2316    2C13  CFMU 031                                       P0=0$K01             FOR BINARY MULT
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

2318   89D6    CFMU 032              CMLT 061  FLTMU2     BR                 BR TO SHARE BINARY MULT
231A   54B2    CFMU 033    LONGEN                         RDH G DA,AE        GET OP CODE
231C   5AF9    CFMU 034                                   H1=T0
231E   5B89    CFMU 035                                   I0=T1              SHIFT
2320   5699    CFMU 036                                   I1=D0               DATA
2322   5359    CFMU 037                                   G1=V1                FOR
2324   5039    CFMU 038                                   V1=U0                 POSTNORMALIZE
2326   5109    CFMU 039                                   U0=U1
2328   5219    CFMU 040                                   U1=V0
232A   5729    CFMU 041                                   V0=D1
232C   46C6    CFMU 042    MULTEN                         D=P                SAVE FLAGS
232F   5CC2    CFMU 043                                   RDH P DA,B8        GET MASK
2330   0FF3    CFMU 044                                   Z=H1*-K0F          CHECK FOR NORMALIZE
2332   C4BC    CFMU 045                    050   ISNORM   BR IF ZNZ
2334   B2F8    CFMU 046              CFC0 034  SHIFL4     BAL                GO NORMALIZE
2336   26FF    CFMU 047                                   D0=D0+KFF          DEC
2338   F4BD    CFMU 048                    050   ISNORM   BR IF AC=1
233A   3713    CFMU 049                                   D1=D1$KC1          SET UNDERFLOW FLAG
233C   5AE2    CFMU 050    ISNORM                         RDH T DA,BC        GET R1 ADD T=00X8
233E   B170    CFMU 051              CFC0 056  ZTEST      BAL
2340   D5C5    CFMU 052                    054   RESNOZ   BR IF S1=1
2342   B1A0    CFMU 053              CFHA 029  DIVNOI     BR                 STORE TRUE 0  REST I  GO TO ICYC
2344   56E9    CFMU 054    RESNOZ                         H0=D0              MOVE RESULT EXP
2346   1E85    CFMU 055                                   H0=H0*-K80         REMOVE SIGN IN CASE OF OVF
2348   C74C    CFMU 056                    058   PLUS     BR IF D10=0        BR IF RES IS +
234A   3E85    CFMU 057                                   H0=H0$K80          MAKE RES -
234C   2080    CFMU 058    PLUS                           SET S4
234E   F300    CFMU 059                    062   NOTUND   BR IF D17=0
2350   27D3    CFMU 060                                   D1=0$K0D
2352   B19E    CFMU 061              CFHA 028  UNDCHK     BR
2300   C605    CFMU 062    NOTUND          064   OVF      BR IF D00=1
2302   0080    CFMU 063                                   RST S4
2304   27C3    CFMU 064    OVF                            D1=0$K0C
2306   B1A6    CFMU 065              CFHA 032  STORE1     BR
                                     **********************************
                                     * CROSS REFERENCE FOR CSECT CFMU *
                                     **********************************

CFMU 009    CMLT 064
CFMU 024    CFMU 010
CFMU 033    CFMU 025
CFMU 042    CFMU 023
CFMU 050    CFMU 045  CFMU 048
CFMU 054    CFMU 052
CFMU 058    CFMU 056
CFMU 062    CFMU 059
CFMU 064    CFMU 062
```

THE I-CYCLES ROUTINE IS ENTERED AT FIVE MAJOR ENTRY POINTS.

LABEL CHECKX - ENTRY IS MADE HERE FROM MICROROUTINES THAT NEED CONDITION CODE ZERO SET.

LABEL CHECK - ENTRY HERE IS FROM ROUTINES THAT DID NOT BRANCH OUT TESTING FOR EXECUTE AND INTERRUPT.

LABEL ISTARTO THIS BRANCH SET ENTRY IS MADE FROM ROUTINES THAT TESTED FOR EXECUTE AND INTERRUPT AS THE EXIT FROM THAT ROUTINE WAS MADE.

LABEL ADDERR - ENTRY AT THIS POINT IS MADE WHEN AN ADDRESSING ERROR HAS BEEN DETECTED. THE ERROR IS FLAGGED AND THE *BPSW* ROUTINE IS BRANCHED TO. THE *BPSW* ROUTINE STORES THE OLD PSW, INCLUDING THE INTERRUPT CODE THAT INDICATES AN ADDRESSING ERROR.

LABEL SPECHK - ENTRY IS MADE HERE FROM ROUTINES IN WHICH A SPECIFICATION EXCEPTION IS DETECTED. THE SPECIFICATION EXCEPTION IS FLAGGED AND A BRANCH IS TAKEN TO THE *BPSW* ROUTINE. THE *BPSW* ROUTINE STORES THE OLD PSW, INCLUDING THE INTERRUPTION CODE THAT INDICATES A SPECIFICATION EXCEPTION.

THE ENTRY POINTS CHECKX, CHECK, AND ISTART ARE THE NORMAL ENTRY POINTS TO THE I-CYCLES ROUTINE.

IF THE ENTRY IS MADE FROM AN EXECUTE INSTRUCTION, THE SAVED INSTRUCTION ADDRESS IS RESTORED AND THE NEW INSTRUCTION READ OUT BEGINS.

IF THE ENTRY IS MADE FROM AN EXECUTE INSTRUCTION, AND AN INTERRUPT IS PENDING, THE SAVED INSTRUCTION ADDRESS IS RESTORED AND A BRANCH TO THE *BSWI* ROUTINE IS MADE. THE *BSWI* ROUTINE WILL DETERMINE THE TYPE OF INTERRUPT AND HANDLE IT.

IF THE ENTRY IS MADE WITH ONLY AN INTERRUPT PENDING, THE *BPSW* ROUTINE IS BRANCHED TO DIRECTLY.

IF THE ENTRY IS MADE WITH NO INTERRUPT PENDING AND NO EXECUTE, THE FIRST 2 INSTRUCTION BYTES ARE READ OUT AND A TEST IS MADE ON THE HIGH 4 BITS OF THE OP CODE.

TEST HIGH 4 BITS OF OP CODE

HIGH HEX DIGIT	OPERATION
0,1	TEST LOW 4 BITS OF OP CODE AND BRANCH TO THE PROPER ROUTINE.
2,3,6,7	THESE ARE FLOATING POINT OPS. A BRANCH IS TAKEN DIRECTLY TO THE *CFCY* ROUTINE. THIS IS THE FLOATING POINT I-CYCLES ROUTINE.
4,5	READ 2ND 2 BYTES OF INSTRUCTION , FIND THE EFFECTIVE OPERAND ADDRESS, TEST LOW 4 BITS OF OP CODE, BRANCH TO PROPER ROUTINE.
8	READ 2ND 2 BYTES OF INSTRUCTION, FIND THE EFFECTIVE OPERAND ADDRESS. TEST BIT 4 OF OP CODE, IF A SHIFT OP, BRANCH TO *CSFT*, IF NOT A SHIFT, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
9	FIND EFFECTIVE OPERAND ADDRESS, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
A,B,C,E	INCREMENT INSTRUCTION ADDRESS +2 FOR A AND B, +4 FOR C AND E. FLAG INVALID OP AND BRANCH TO *BPSW* ROUTINE WHERE THE OPERATION INTERRUPTION CODE IS STORED IN THE OLD PSW.
D	FIND EFFECTIVE OPERAND ADDRESSES, TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.
F	FIND EFFECTIVE OPERAND ADDRESSES, ADD IN THE OPERAND LENGTH CODES TO THE EFFECTIVE ADDRESSES TO POINT TO LOW ORDER BYTES. TEST LOW 4 BITS OF OP CODE AND BRANCH TO PROPER ROUTINE.

ADDR WORD SEQUENCE NO. LABEL NEXTSEQ NEXTLABEL STATEMENT COMMENTS

```
                CICY 001    T              INSTRUCTION CYCLES 360 OP SET                    001
                CICY 002    *
```

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2A74	2C07	CICY 003	CHECKX			P0=0	SET CC TO EQUAL
2A76	8185	CICY 004	CHECK	005	ISTART N	N=S BITS67	TEST FOR EXECUTE OR INTERRUPT
2A00	A99C	CICY 005	ISTART 0	010	EXCHK	BR	EXECUTE AND INT
2A02	A99C	CICY 006	ISTART 1	010	EXCHK	BR	EXECUTE NOT INT
2A04	8446	CICY 007	ISTART 2	BSWI 007	CHECKB	BR	INT NOT EXECUTE
2A06	5498	CICY 008	ISTART 3			RDH G I+2	READ FIRST TWO INST. BYTES
2A08	B463	CICY 009	OPHIBR	013	OPHI N	N=GOH	TEST HIGH OP CODE DIGIT
299C	5892	CICY 010	EXCHK			RDH I DA,AA	READ SAVED ADDRESS
299E	2020	CICY 011				SET S6	TURN OFF EXECUTE
29A0	8185	CICY 012		005	ISTART N	N=S BITS67	
2960	B053	CICY 013	OPHI 0	133	OPOX N	N=GOL	RR OP
2962	B073	CICY 014	OPHI 1	169	OP1X N	N=GOL	CODE BREAKDOWN
2964	A8C4	CICY 015	OPHI 2	CFCY 012	RROP	BR	FLOATING POINT OPS
2966	A8C4	CICY 016	OPHI 3	CFCY 012	RROP	BR	FLOATING POINT OPS
2968	8100	CICY 017	OPHI 4	064	OPHI4	BR	RX OP
296A	AE8C	CICY 018	OPHI 5	066	OPHI5	BR	CODE BASE CALCULATE
296C	A8A0	CICY 019	OPHI 6	CFCY 034	RXOP	BR	FLOATING POINT OPS
296E	A8A0	CICY 020	OPHI 7	CFCY 034	RXOP	BR	FLOATING POINT OPS
2970	9044	CICY 021	OPHI 8	058	OPHI8	BR	RS OPS
2972	8108	CICY 022	OPHI 9	062	OPHI9	BR	SI OPS
2974	83B2	CICY 023	OPHI A	076	RSINV	BR	INVALID
2976	83B2	CICY 024	OPHI B	076	RSINV	BR	OP
2978	83B0	CICY 025	OPHI C	075	SSINV	BR	CODES
297A	8104	CICY 026	OPHI D	073	OPHID	BR	SS OPS
297C	83B0	CICY 029	OPHI E	075	SSINV	BR	INVALID OPS
297E	8EC8	CICY 034	OPHI F	043	SSFX	BR	DECIMAL OPS
0EC8	814C	CICY 043	SSFX	111	BASE3	BAL	
0ECA	55E5	CICY 044				H0=G1XL	CALCULATE
0ECC	61EB	CICY 045				U1C=U1+H0	RIGHT MOST
0ECE	60AD	CICY 046				U0C=U0+T0+C	ADDRESS
0ED0	6FAD	CICY 047				H1C=H1+T0+C	ADD IN LENGTH CODES TO
0ED2	55ED	CICY 048				H0=G1L	POINT TO LOW BYTE
0ED4	63EB	CICY 049				V1C=V1+H0	OF OPERANDS
0ED6	62AD	CICY 050				V0C=V0+T0+C	
0ED8	67AD	CICY 051				D1C=D1+T0+C	
0EDA	6F75	CICY 052				H1=H1$D1	
0EDC	AB77	CICY 053		055	ADERR N	BR IF H1=NZ	SS FXOP BREAKOUT
0EDE	B07D	CICY 054		267	OPFX N	N=GOL	FLAG ADDRESS ERR
0BE0	AA20	CICY 055	ADERR 0	056	ADDERR 0	BR	MAKE ADDRESSABLE
2A20	2B53	CICY 056	ADDERR 0			T1=0$K05	FLAG ADDRESS ERROR
2A22	89FC	CICY 057		BPSW 003	PROG	BR	GO TAKE INTERUPT
1044	834A	CICY 058	OPHI8	081	BASE1	BAL	RS OP CODE 8X
1046	C043	CICY 059		061	ALLSHI	BR IF G04=1	SHIFT OPS
1048	9023	CICY 060		217	OP8X N	N=GO BITS567	8X OPS LOW DIGIT TEST
1042	9CFA	CICY 061	ALLSHI	CSFT 014	SHIFT	BR	SHIFT OPS EXIT
0108	834A	CICY 062	OPHI9	081	BASE1	BAL	RS OP CODE 9X
010A	B009	CICY 063		225	OP9X N	N=GOL	BREAKOUT
0100	A6D8	CICY 064	OPHI4	092	BASE2	BAL	RX GO TO BASE + INDEX
0102	B055	CICY 065		185	OP4X N	N=GOL	BREAKOUT
2E8C	A6D8	CICY 066	OPHI5	092	BASE2	BAL	RX GO TO BASE AND INDEX
2E8E	AB25	CICY 067		056	ADDERR N	BR IF H1=NZ	CHECK FOR ADDRESS ERROR

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL   STATEMENT            COMMENTS
2E90   01C3    CICY 068                                    Z=U1*-K0C            TEST FOR
2E92   F08B    CICY 069                 072     OPHI5A     BR IF LZ=0             SPECIFICATION CHECK
2E94   2B63    CICY 070    SPECHK                          T1=0$K06             FLAG SPEC CHECK
2E96   89FC    CICY 071                BPSW 003 PROG       BR                   BRANCH TO STORE INTERRUPT CODE
2E8A   B071    CICY 072    OPHI5A       201     OP5X   N   N=G0L                RX 5XOP BREAKOUT
0104   814C    CICY 073    OPHID        111     BASE3      BAL                  SS DXOP
0106   B075    CICY 074                 251     OPDX   N   N=G0L                 BREAKOUT
03B0   6884    CICY 075    SSINV                           I=I+2                INCREMENT
03B2   6884    CICY 076    RSINV                           I=I+2                 INSTR ADDRESS
03B4   2B13    CICY 077    OPERR                           T1=0$K01               AND FLAG ERROR
03B6   89FC    CICY 078                BPSW 003 PROG       BR                   BRANCH TO STORE INTERRUPT CODE
2DFC   2B23    CICY 079    PRIVOP                          T1=0$K02              FLAG PRIV OP
2DFE   89FC    CICY 080                BPSW 003 PROG       BR                   BRANCH TO STORE INTERRUPT CODE
034A   5098    CICY 081    BASE1                           RDH U I+2            READ 2ND 2 INST BYTES
034C   50BB    CICY 082                                    T1=U0H                AND PUT IN REG. ADD.
034E   C4C7    CICY 083                 090     OUT1       BR IF Z=0              BRANCH IF NO BASE
0350   5EA8    CICY 084                                    RDH H AS,T+2         GET BASE DATA
0352   56A0    CICY 085                                    RDH D AS,T            AND
0354   617B    CICY 086                                    U1C=U1+D1               CALCULATE
0356   606F    CICY 087                                    U0C=U0L+D0+C               EFFECTIVE
0358   6FAD    CICY 088                                    H1C=H1+T0+C                   ADDRESS
035A   128E    CICY 089                                    RTN
0346   2F07    CICY 090    OUT1                            H1=0                 INDICATE ZERO BASE
0348   128E    CICY 091                                    RTN
26D8   5098    CICY 092    BASE2                           RDH U I+2            READ 2ND 2 INST. BYTES
26DA   50BB    CICY 093                                    T1=U0H
26DC   C4F9    CICY 094                 108     BASE2A     BR IF Z=0            NO BASE  BR
26DE   5EA8    CICY 095                                    RDH H AS,T+2         GET
26E0   56A0    CICY 096                                    RDH D AS,T            REGISTER DATA
26E2   617B    CICY 097                                    U1C=U1+D1            CALCULATE
26E4   606F    CICY 098                                    U0C=U0L+D0+C          BASE
26E6   6FAD    CICY 099                                    H1C=H1+T0+C            VALUE
26E8   55B3    CICY 100    BASE2B                          T1=G1XH              GET INDEX REG.
26EA   C4F7    CICY 101                 107     BASE2C     BR IF Z=0            NO INDEX BR
26EC   52A8    CICY 102                                    RDH V AS,T+2         GET
26EE   56A0    CICY 103                                    RDH D AS,T            REGISTER DATA
26F0   617B    CICY 104                                    U1C=U1+D1            CALCULATE
26F2   606D    CICY 105                                    U0C=U0+D0+C           INDEX
26F4   6F3D    CICY 106                                    H1C=H1+V1+C            VALUE
26F6   128E    CICY 107    BASE2C                          RTN
26F8   2F07    CICY 108    BASE2A                          H1=0                 INDICATE ZERO BASE
26FA   C4E9    CICY 109                 100     BASE2B     BR IF Z=0            UNCONDITIONAL BRANCH
014C   5098    CICY 111    BASE3                           RDH U I+2            READ 2ND 2 INST. BYTES
014E   50BB    CICY 112                                    T1=U0H               GET BASE ADDRESS
0150   C4C5    CICY 113                 129     BASE3A     BR IF Z=0            BRANCH IF NO BASE 1
0152   5EA8    CICY 114                                    RDH H AS,T+2         GET
0154   56A0    CICY 115                                    RDH D AS,T            BASE DATA
0156   617B    CICY 116                                    U1C=U1+D1            CALCULATE
0158   606F    CICY 117                                    U0C=U0L+D0+C          1ST
015A   6FAD    CICY 118                                    H1C=H1+T0+C            OPERAND
015C   5298    CICY 119    BASE3B                          RDH V I+2            READ 3RD 2 INST BYTES
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
015E    52BB    CICY 120                                     T1=VOH
0160    C4C9    CICY 121                131       BASE3C     BR IF Z=0          BRANCH IF NO BASE 2
0162    6AA4    CICY 122                                     T=T+2
0164    56AA    CICY 123                                     RDH D AS,T-2
0166    637B    CICY 124                                     V1C=V1+D1          CALCULATE
0168    626F    CICY 125                                     VOC=VOL+DO+C        2ND
016A    56AO    CICY 126                                     RDH D AS,T           OPERAND
016C    67AD    CICY 127                                     D1C=D1+TO+C
016E    128E    CICY 128                                     RTN
0144    2F07    CICY 129    BASE3A                           H1=0               INDICATE ZERO BASE
0146    C4DD    CICY 130                119       BASE3B     BR IF Z=0          UNCONDITIONAL BRANCH
0148    2707    CICY 131    BASE3C                           D1=0               INDICATE ZERO BASE
014A    128E    CICY 132                                     RTN
29C0    83B4    CICY 133    OPOX  0      077       OPERR      BR                   *
29C2    83B4    CICY 134    OPOX  1      077       OPERR      BR                   * INVALID OP CODES
29C4    83B4    CICY 135    OPOX  2      077       OPERR      BR                   *
29C6    83B4    CICY 136    OPOX  3      077       OPERR      BR                   *
29C8    AE32    CICY 137    OPOX  4  CSTS 087      RROP04     BR                 SET PROGRAM MASK
29CA    A912    CICY 138    OPOX  5  CBRC 023      RROP05     BR                 BRANCH AND LINK RR
29CC    ADA8    CICY 139    OPOX  6  CBRC 068      RROP06     BR                 BRANCH ON COUNT RR
29CE    A93A    CICY 140    OPOX  7  CBRC 005      RROP07     BR                 BRANCH CONDITION RR
29D0    AA34    CICY 141    OPOX  8  CSTS 044      RROP08     BR                 SET KEY
29D2    AA34    CICY 142    OPOX  9  CSTS 044      RROP08     BR                 INSERT KEY
29D4    B364    CICY 143    OPOX  A  CSTS 098      RROPOA     BR                 SUPERVISOR CALL
29D6    83B4    CICY 146    OPOX  B      077       OPERR      BR                   *
29D8    83B4    CICY 164    OPOX  C      077       OPERR      BR                   *
29DA    83B4    CICY 165    OPOX  D      077       OPERR      BR                   * INVALID OP CODES
29DC    83B4    CICY 166    OPOX  E      077       OPERR      BR                   *
29DE    83B4    CICY 167    OPOX  F      077       OPERR      BR                   *
29E0    AC34    CICY 169    OP1X  0  CLST 005      RROP10     BR               LOAD POSITIVE
29E2    AC28    CICY 170    OP1X  1  CLST 030      RROP11     BR               LOAD NEGATIVE
29E4    AC0C    CICY 171    OP1X  2  CLST 018      RROP12     BR               LOAD AND TEST
29E6    AC00    CICY 172    OP1X  3  CLST 040      RROP13     BR               LOAD COMPLEMENT
29E8    B480    CICY 173    OP1X  4  CLOG 078      RROP14     BR               AND
29EA    B27A    CICY 174    OP1X  5  CLOG 047      RROP15     BR               COMPARE
29EC    A57C    CICY 175    OP1X  6  CLOG 114      RROP16     BR               OR
29EE    B48C    CICY 176    OP1X  7  CLOG 144      RROP17     BR               EXCLUSIVE OR
29F0    AE98    CICY 177    OP1X  8  CLST 046      RROP18     BR               LOAD
29F2    94C2    CICY 178    OP1X  9  CBIN 005      RROP19     BR               COMPARE
29F4    A4FC    CICY 179    OP1X  A  CBIN 022      RROP1A     BR               ADD
29F6    A77C    CICY 180    OP1X  B  CBIN 038      RROP1B     BR               SUBTRACT
29F8    A06C    CICY 181    OP1X  C  CMLT 007      RROP1C     BR               MULTIPLY
29FA    8390    CICY 182    OP1X  D  CDVD 002      RROP1D     BR               DIVIDE
29FC    AE78    CICY 183    OP1X  E  CBIN 051      RROP1E     BR               ADD LOGICAL
29FE    A078    CICY 184    OP1X  F  CBIN 072      RROP1F     BR               SUBTRACT LOGICAL
02C0    ACD4    CICY 185    OP4X  0  CLST 055      RXOP40     BR               SUBTRACT HALFWORD
02C2    AD76    CICY 186    OP4X  1  CLOG 200      RXOP41     BR               LOAD ADDRESS
02C4    AEA6    CICY 187    OP4X  2  CLOG 192      RXOP42     BR               STORE CHARACTER
02C6    A874    CICY 188    OP4X  3  CLOG 184      RXOP43     BR               INSERT CHARACTER
02C8    AD88    CICY 189    OP4X  4  CBRC 047      RXOP44     BR               EXECUTE          RX
02CA    A91C    CICY 190    OP4X  5  CBRC 031      RXOP45     BR               BRANCH AND LINK  RX
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT      COMMENTS
02CC   ADB2    CICY 191    OP4X   6  CBRC 074  RXOP46     BR             BRANCH ON COUNT RX
02CE   A942    CICY 192    OP4X   7  CBRC 011  RXOP47     BR             BRANCH ON CONDITION
02D0   ACC0    CICY 193    OP4X   8  CLST 066  RXOP48     BR             LOAD HALFWORD
02D2   A3EE    CICY 194    OP4X   9  CBIN 018  RXOP49     BR             COMPARE HALFWORD
02D4   88E8    CICY 195    OP4X   A  CBIN 034  RXOP4A     BR             ADD HALFWORD
02D6   969C    CICY 196    OP4X   B  CBIN 047  RXOP4B     BR             SUBTRACT HALFWORD
02D8   8978    CICY 197    OP4X   C  CMLT 014  RXOP4C     BR             MULTIPLY HALFWORD
02DA   83B4    CICY 198    OP4X   D        077  OPERR      BR               * INVALID OP CODE
02DC   AC4E    CICY 199    OP4X   E  CNVR 005  RXOP4E     BR             CONVERT TO DECIMAL
02DE   AC4E    CICY 200    OP4X   F  CNVR 005  RXOP4E     BR             CONVERT TO BINARY
28E0   AEBE    CICY 201    OP5X   0  CLST 078  RXOP50     BR             STORE
28E2   83B4    CICY 202    OP5X   1        077  OPERR      BR               *
28E4   83B4    CICY 203    OP5X   2        077  OPERR      BR               * INVALID OP CODES
28E6   83B4    CICY 204    OP5X   3        077  OPERR      BR               *
28E8   A83A    CICY 205    OP5X   4  CLOG 082  RXOP54     BR             AND
28EA   A86A    CICY 206    OP5X   5  CLOG 050  RXOP55     BR             COMPARE LOGICAL
28EC   8CE4    CICY 207    OP5X   6  CLOG 118  RXOP56     BR             OR
28EE   8114    CICY 208    OP5X   7  CLOG 148  RXOP57     BR             EXCLUSIVE OR
28F0   AEB2    CICY 209    OP5X   8  CLST 086  RXOP58     BR             LOAD
28F2   AD52    CICY 210    OP5X   9  CBIN 008  RXOP59     BR             COMPARE
28F4   AC18    CICY 211    OP5X   A  CBIN 026  RXOP5A     BR             ADD
28F6   88DA    CICY 212    OP5X   B  CBIN 042  RXOP5B     BR             SUBTRACT
28F8   A064    CICY 213    OP5X   C  CMLT 002  RXOP5C     BR             MULTIPLY
28FA   B28C    CICY 214    OP5X   D  CDVD 006  RXOP5D     BR             DIVIDE
28FC   AD42    CICY 215    OP5X   E  CBIN 055  RXOP5E     BR             ADD LOGICAL
28FE   8E7A    CICY 216    OP5X   F  CBIN 076  RXOP5F     BR             SUBTRACT LOGICAL
1120   AA68    CICY 217    OP8X   0  CSTS 011  RSOP80     BR             SET SYSTEM MASK
1122   83B4    CICY 218    OP8X   1        077  OPERR      BR               * INVALID OP CODE
1124   AA24    CICY 219    OP8X   2  CSTS 003  RSOP82     BR             LOAD PSW
1126   AA30    CICY 220    OP8X   3  CSTS 019  RSOP83     BR             DIAGNOSE
1128   8898    CICY 221    OP8X   4  CSTS 112  RSOP84     BR             WRITE DIRECT
112A   8A5E    CICY 222    OP8X   5  CSTS 113  RSOP85     BR             READ DIRECT
112C   AC88    CICY 223    OP8X   6  CBRC 091  RSOP86     BR             BRANCH ON INDEX HIGH
112E   AC88    CICY 224    OP8X   7  CBRC 091  RSOP86     BR             BRANCH ON INDEX LOW OR EQUAL
0400   B412    CICY 225    OP9X   0  CLST 094  RSOP90     BR             STORE MULTIPLE
0402   A85A    CICY 226    OP9X   1  CLOG 174  RSOP91     BR             TEST UNDER MASK
0404   A9BA    CICY 227    OP9X   2  CLOG 004  RSOP92     BR             MOVE
0406   AE44    CICY 228    OP9X   3  CSTS 076  RSOP93     BR             TEST AND SET
0408   A84C    CICY 229    OP9X   4  CLOG 093  RSOP94     BR             AND
040A   A816    CICY 230    OP9X   5  CLOG 056  RSOP95     BR             COMPARE LOGICAL
040C   AB72    CICY 231    OP9X   6  CLOG 126  RSOP96     BR             OR
040E   A958    CICY 232    OP9X   7  CLOG 156  RSOP97     BR             EXCLUSIVE OR
0410   B420    CICY 233    OP9X   8  CLST 103  RSOP98     BR             LOAD MULTIPLE
0412   83B4    CICY 239    OP9X   9        077  OPERR      BR               *
0414   83B4    CICY 241    OP9X   A        077  OPERR      BR               * INVALID OP CODES
0416   83B4    CICY 246    OP9X   B        077  OPERR      BR               *
0418   8886    CICY 247    OP9X   C  DCLA 002  IOINST     BR             START I/O
041A   8886    CICY 248    OP9X   D  DCLA 002  IOINST     BR             TEST I/O
041C   8886    CICY 249    OP9X   E  DCLA 002  IOINST     BR             HALT I/O
041E   8886    CICY 250    OP9X   F  DCLA 002  IOINST     BR             TEST CHANNEL
02E0   83B4    CICY 251    OPDX   0        077  OPERR      BR               * INVALID OP CODE
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT        COMMENTS
02E2     AE56    CICY 252    OPDX   1  CLOG 016  SSOPD1     BR               MOVE NUMERIC
02E4     AE7C    CICY 253    OPDX   2  CLOG 008  SSOPD2     BR               MOVE
02E6     AE20    CICY 254    OPDX   3  CLOG 026  SSOPD3     BR               MOVE ZONES
02E8     A944    CICY 255    OPDX   4  CLOG 102  SSOPD4     BR               AND
02EA     A828    CICY 256    OPDX   5  CLOG 067  SSOPD5     BR               COMPARE LOGICAL
02EC     A988    CICY 257    OPDX   6  CLOG 132  SSOPD6     BR               OR
02EE     AD62    CICY 258    OPDX   7  CLOG 162  SSOPD7     BR               EXCLUSIVE OR
02F0     83B4    CICY 259    OPDX   8       077  OPERR      BR                 *
02F2     83B4    CICY 260    OPDX   9       077  OPERR      BR                 * INVALID OP CODES
02F4     83B4    CICY 261    OPDX   A       077  OPERR      BR                 *
02F6     83B4    CICY 262    OPDX   B       077  OPERR      BR                 *
02F8     ADDO    CICY 263    OPDX   C  CTRT 003  SSOPDC     BR               TRANSLATE
02FA     ADDC    CICY 264    OPDX   D  CTRT 003  SSOPDC     BR               TRANSLATE AND TEST
02FC     AA78    CICY 265    OPDX   E  CTRT 033  SSOPDE     BR               EDIT
02FE     AA78    CICY 266    OPDX   F  CTRT 033  SSOPDE     BR               EDIT AND MARK
0EE0     83B4    CICY 267    OPFX   0       077  OPERR      BR                 * INVALID OP CODE
0EE2     8136    CICY 268    OPFX   1  CMPU 003  SSOPF1     BR               MOVE WITH OFFSET
0EE4     AE6E    CICY 269    OPFX   2  CMPU 011  SSOPF2     BR               PACK
0EE6     AE68    CICY 270    OPFX   3  CMPU 008  SSOPF3     BR               UNPACK
0EE8     83B4    CICY 272    OPFX   4       077  OPERR      BR                 *
0EEA     83B4    CICY 273    OPFX   5       077  OPERR      BR                 * INVALID OP CODES
0EEC     83B4    CICY 278    OPFX   6       077  OPERR      BR                 *
0EEE     83B4    CICY 279    OPFX   7       077  OPERR      BR                 *
0EF0     934C    CICY 280    OPFX   8  CSAS C06  SSOPF8     BR               ZERO AND ADD
0EF2     93BA    CICY 281    OPFX   9  CSAS 061  SSOPF9     BR               COMPARE DECIMAL
0EF4     934E    CICY 282    OPFX   A  CSAS 007  SSOPFA     BR               ADD DECIMAL
0EF6     93B8    CICY 283    OPFX   B  CSAS 060  SSOPFB     BR               SUBTRACT DECIMAL
0EF8     97A8    CICY 284    OPFX   C  CDMD 002  SSOPFC     BR               MULTIPLY DECIMAL
0EFA     97A8    CICY 285    OPFX   D  CDMD 002  SSOPFC     BR               DIVIDE DECIMAL
0EFC     83B4    CICY 287    OPFX   E       077  OPERR      BR                 *
0EFE     83B4    CICY 288    OPFX   F       077  OPERR      BR                 * INVALID OP CODES
                                                ************************************
                                                * CROSS REFERENCE FOR CSECT CICY *
                                                ************************************
CICY 003    CCOM 156  CFAD 075  CSAS 099  CSFT 084  CTRT 021  CTRT 060
CICY 004    CDMD 149  CDVD 087  CFAD 130  CFHA 035  CFLS 043  CMLT 083  CSAS 128  CSFT 066  CSTS 070  DCHN 126
CICY 005    BSWI 115  CBIN 015  CBIN 062  CBIN 065  CBIN 067  CBIN 069  CBRC 013  CBRC 018  CBRC 020  CBRC 043  CBRC 086  CBRC 118
            CCOM 160  CCOM 164  CCOM 168  CICY 004  CICY 012  CLOG 006  CLOG 014  CLOG 024  CLOG 034  CLOG 062  CLOG 064  CLOG 099
            CLOG 181  CLOG 189  CLOG 197  CLOG 204  CLST 013  CLST 027  CLST 052  CLST 062  CLST 083  CLST 091  CMPU 018  CMPU 019
            CNVR 056  CNVR 122  CSTS 016  CSTS 059  CSTS 082  CSTS 084  CSTS 095  CTRT 016
CICY 008    BPSW 063  CBRC 066
CICY 010    CICY 005  CICY 006
CICY 013    CBRC 062  CICY 009
CICY 043    CICY 034
CICY 055    CICY 053
CICY 056    BWRP 120  BWRP 122  CBRC 047  CCOM 012  CFCY 039  CICY 055  CICY 067  CLOG 004  CLOG 009  CLOG 017  CLOG 027  CLOG 056
            CLOG 068  CLOG 093  CLOG 103  CLOG 126  CLOG 133  CLOG 156  CLOG 163  CLOG 174  CLOG 184  CLOG 192  CLST 055  CLST 066
            CLST 110  CNVR 005  CSTS 003  CSTS 011  CSTS 047  CSTS 076  CTRT 003  CTRT 010  CTRT 033  CTRT 066
CICY 058    CICY 021
CICY 061    CICY 059
CICY 062    CICY 022
```

****
* CROSS REFERENCE FOR CSECT CICY *
****

```
CICY 064     CICY 017
CICY 066     CICY 018
CICY 070     CBRC 065  CCOM 027  CDMD 005  CDVD 010  CFCY 051  CLST 063  CMLT 006  CNVR 008  CSFT 023  CSTS 009
CICY 072     CICY 069
CICY 073     CICY 026
CICY 075     CICY 025  CICY 029
CICY 076     CICY 023  CICY 024
CICY 077     CFCY 069  CICY 133  CICY 134  CICY 135  CICY 136  CICY 146  CICY 164  CICY 165  CICY 166  CICY 167  CICY 198  CICY 202
             CICY 203  CICY 204  CICY 218  CICY 239  CICY 241  CICY 246  CICY 251  CICY 259  CICY 260  CICY 261  CICY 262  CICY 267
             CICY 272  CICY 273  CICY 278  CICY 279  CICY 287  CICY 288  CSTS 112  CSTS 113
CICY 079     CSTS 021  DCLA 004
CICY 081     CICY 058  CICY 062
CICY 090     CICY 083
CICY 092     CFCY 034  CICY 064  CICY 066
CICY 100     CICY 109
CICY 107     CICY 101
CICY 108     CICY 094
CICY 111     CICY 043  CICY 073
CICY 119     CICY 130
CICY 129     CICY 113
CICY 131     CICY 121
CICY 133     CICY 013
CICY 169     CICY 014
CICY 185     CICY 065
CICY 201     CICY 072
CICY 217     CICY 060
CICY 225     CICY 063
CICY 251     CICY 074
CICY 267     CICY 054
```

CLOG -- LOGICAL OPERATIONS

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP14	AND	14	R1 R2		
RROP15	COMPARE	15	R1 R2		
RROP16	OR	16	R1 R2		
RROP17	EXCLUSIVE OR	17	R1 R2		
RXOP41	LOAD ADDRESS	41	R1 X2	B2 D2	D2
RXOP42	STORE CHARACTER	42	R1 X2	B2 D2	D2
RXOP43	INSERT CHARACTER	43	R1 X2	B2 D2	D2
RXOP54	AND	54	R1 X2	B2 D2	D2
RXOP55	COMPARE LOGICAL	55	R1 X2	B2 D2	D2
RXOP56	OR	56	R1 X2	B2 D2	D2
RXOP57	EXCLUSIVE OR	57	R1 X2	B2 D2	D2
RSOP91	TEST UNDER MASK	91	I2	B1 D1	D1
RSOP92	MOVE	92	I2	B1 D1	D1
RSOP94	AND	94	I2	B1 D1	D1
RSOP95	COMPARE LOGICAL	95	I2	B1 D1	D1
RSOP96	OR	96	I2	B1 D1	D1
RSOP97	EXCLUSIVE OR	97	I2	B1 D1	D1

SS FORMAT INSTRUCTIONS

OP CODE.	L1	L2	B1	D1	D1	D1	B2	D2	D2	D2

ENTRY POINTS	OPERATION
SSOPD1	MOVE NUMERIC
SSOPD2	MOVE
SSOPD3	MOVE ZONES
SSOPD4	AND
SSOPD5	COMPARE LOGICAL
SSOPD6	OR
SSOPD7	EXCLUSIVE OR
SSOPE8	MOVE INVERSE

OBJECTIVES

AND, OR, EXCLUSIVE OR
THE LOGIC FUNCTION IS PERFORMED WITH CORRESPONDING BITS IN THE 1ST AND 2ND OPERANDS. THE RESULT REPLACES THE 1ST OPERAND. IN THE SI FORMAT, I2 IS THE 2ND OPERAND AND IS ONE BYTE LONG.

COMPARE
COMPARE 1ST OPERAND TO THE 2ND OPERAND. OPERANDS ARE TREATED AS UNSIGNED BINARY QUANTITIES.

STORE CHARACTER
MOVE BYTE 3 OF R1 TO PROGRAM STORAGE ADDRESSED BY 2ND OPERAND.

MOVE
THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION. IN MOVE IMMEDIATE, I2 IS THE 2ND OPERAND.

MOVE NUMERIC
SAME AS MOVE, EXCEPT ONLY THE LOW-ORDER 4 BITS ARE MOVED.

MOVE ZONES
SAME AS MOVE, EXCEPT ONLY THE HIGH-ORDER 4 BITS ARE MOVED.

TEST UNDER MASK
THE IMMEDIATE DATA IN BYTE 2 OF INSTRUCTION IS TESTED BIT FOR BIT AGAINST THE 1ST OPERAND.

LOAD ADDRESS
SET THE EFFECTIVE ADDRESS OF THE 2ND OPERAND INTO THE LOW 24 BITS OF R1.

INVERSE MOVE (RPQ)
THE LOW ORDER POSITION OF THE SECOND OPERAND IS MOVED TO THE HIGH ORDER POSITION OF THE FIRST OPERAND.MODIFYING THE FIRST OPERAND ADDRESS WITH +1 AND THE SECOND OPERAND ADDRESS WITH -1,THE OPERATION IS TERMINATED WITH LENGTH COUNT ZERO.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CLOG 001    T'          LOGICAL OPERATIONS
              CLOG 002    *
              CLOG 003    *           MOVE IMMEDIATE
29BA    AB25  CLOG 004     RSOP92   CICY 056  ADDERR N  BR IF H1=NZ        MOVE
29BC    7510  CLOG 005                                  STB G1 U             IMMEDIATE
29BE    8185  CLOG 006              CICY 005  ISTART N  N=S BITS67           EXECUTION
              CLOG 007    *           MOVE CHARACTERS
2E7C    6F75  CLOG 008     SSOPD2                       H1=H1$D1             MOVE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
2E7E    AB25    CLOG 009              CICY 056  ADDERR N   BR IF H1=NZ          CHARACTERS
2E80    5738    CLOG 010    OPD2                           RDB D1 V+1            EXECUTION
2E82    7718    CLOG 011                                   STB D1 U+1
2E84    25FF    CLOG 012                                   G1=G1+KFF
2E86    F481    CLOG 013                  010  OPD2        BR IF AC=1
2E88    8185    CLOG 014              CICY 005  ISTART N   N=S BITS67
                CLOG 015  *             MOVE NUMERICS
2E56    6F75    CLOG 016    SSOPD1                         H1=H1$D1          MOVE
2E58    AB25    CLOG 017              CICY 056  ADDERR N   BR IF H1=NZ          NUMERICS
2E5A    5738    CLOG 018    OPD1                           RDB D1 V+1            EXECUTION
2E5C    5F10    CLOG 019                                   RDB H1 U
2E5E    4F78    CLOG 020                                   D1=H1H+D1L
2E60    7718    CLOG 021                                   STB D1 U+1
2E62    25FF    CLOG 022                                   G1=G1+KFF
2E64    F4DB    CLOG 023                  018  OPD1        BR IF AC=1
2E66    8185    CLOG 024              CICY 005  ISTART N   N=S BITS67
                CLOG 025  *             MOVE ZONES
2E20    6F75    CLOG 026    SSOPD3                         H1=H1$D1          MOVE
2E22    AB25    CLOG 027              CICY 056  ADDERR N   BR IF H1=NZ          ZONES
2E24    5738    CLOG 028    OPD3                           RDB D1 V+1            EXECUTION
2E26    5F10    CLOG 029                                   RDB H1 U
2E28    4F7D    CLOG 030                                   D1=H1L+D1H
2E2A    7718    CLOG 031                                   STB D1 U+1
2E2C    25FF    CLOG 032                                   G1=G1+KFF
2E2E    F4A5    CLOG 033                  028  OPD3        BR IF AC=1
2E30    8185    CLOG 034              CICY 005  ISTART N   N=S BITS67
                CLOG 046  *             COMPARE LOGICAL    RR                          COMPARE
327A    8270    CLOG 047    RROP15    CCOM 036  CRGET       BAL                       LOGICAL
327C    A86C    CLOG 048                  051  CXCOML      BR                         INSTRUCTIONS
                CLOG 049  *             COMPARE LOGICAL    RX                                EXECUTION
286A    B3F6    CLOG 050    RXOP55    CCOM 045  CXGET       BAL
286C    B460    CLOG 051    CXCOML    CCOM 086  CSLOOP      BAL
286E    D5D6    CLOG 052                  098  OP94B       BR IF S1=0          GO SET CC 0
2870    F5A5    CLOG 053                  063  OP95A       BR IF S3=1          GO SET CC2
2872    A820    CLOG 054                  061  OP95C  0    BR                  GO SET CC1
                CLOG 055  *             COMPARE LOGICAL RS
2816    AB25    CLOG 056    RSOP95    CICY 056  ADDERR N   BR IF H1=NZ
2818    5F10    CLOG 057                                   RDB H1 U
281A    7F51    CLOG 058                                   H1=H1-G1+1
281C    C4D7    CLOG 059                  098  OP94B       BR IF Z=0
281E    F4A5    CLOG 060    OP95D         063  OP95A       BR IF AC=1
2820    2C55    CLOG 061    OP95C  0                       P0=0$K50            SET CC 1
2822    8185    CLOG 062              CICY 005  ISTART N   N=S BITS67
2824    2C25    CLOG 063    OP95A                          P0=0$K20            SET CC 2
2826    8185    CLOG 064              CICY 005  ISTART N   N=S BITS67
                CLOG 065  *
                CLOG 066  *             COMPARE LOGICAL CHARACTERS
2828    6F75    CLOG 067    SSOPD5                         H1=H1$D1
282A    AB25    CLOG 068              CICY 056  ADDERR N   BR IF H1=NZ
282C    5738    CLOG 069    OPD5                           RDB D1 V+1
282E    5F18    CLOG 070                                   RDB H1 U+1
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS
2830   7F71    CLOG 071                                    H1=H1-D1+1
2832   C49E    CLOG 072                 060    OP95D       BR IF ZNZ
2834   25FF    CLOG 073                                    G1=G1+KFF
2836   F4AD    CLOG 074                 069    OPD5        BR IF AC=1
2838   A856    CLOG 075                 098    OP94B       BR
               CLOG 076   *
               CLOG 077   *           AND   RR
3480   8270    CLOG 078    RROP14    CCOM 036  CRGET       BAL                  AND
3482   A83C    CLOG 079                 083    CXAND       BR                    EXECUTION
               CLOG 080   *
               CLOG 081   *           AND   RX
283A   B3F6    CLOG 082    RXOP54    CCOM 045  CXGET       BAL
283C   6F17    CLOG 083    CXAND                           H1=H1*U1
283E   6E07    CLOG 084                                    H0=H0*U0
2840   6737    CLOG 085                                    D1=D1*V1
2842   6627    CLOG 086                                    D0=D0*V0
2844   76A8    CLOG 087    CCLOST                          STH D AS,T+2
2846   7EAA    CLOG 088                                    STH H AS,T-2
2848   E5A1    CLOG 089    S2CHK                 061    OP95C   0   BR IF S2=1
284A   A856    CLOG 090                 098    OP94B       BR
               CLOG 091   *
               CLOG 092   *           AND   RS
284C   AB25    CLOG 093    RSOP94    CICY 056  ADDERR  N   BR IF H1=NZ
284E   5F10    CLOG 094                                    RDB H1 U
2850   6F57    CLOG 095                                    H1=H1*G1
2852   7F10    CLOG 096    OP94A                           STB H1 U
2854   AB21    CLOG 097                 061    OP95C   N   BR IF H1=NZ
2856   2C07    CLOG 098    OP94B                           P0=0                 SET CCO
2858   8185    CLOG 099               CICY 005  ISTART  N  N=S BITS67
               CLOG 100   *
               CLOG 101   *           AND   SS
2944   6F75    CLOG 102    SSOPD4                          H1=H1$D1
2946   AB25    CLOG 103               CICY 056  ADDERR  N  BR IF H1=NZ
2948   10CE    CLOG 104                                    RST S K=FC
294A   5738    CLOG 105    OPD4                            RDB D1 V+1
294C   5F10    CLOG 106                                    RDB H1 U
294E   67F7    CLOG 107                                    D1=D1*H1
2950   7718    CLOG 108                                    STB D1 U+1
2952   25FF    CLOG 109                                    G1=G1+KFF
2954   F4CB    CLOG 110                 105    OPD4        BR IF AC=1
2956   A848    CLOG 111                 089    S2CHK       BR
               CLOG 112   *
               CLOG 113   *           OR    RR
257C   8270    CLOG 114    RROP16    CCOM 036  CRGET       BAL
257E   8CE6    CLOG 115                 119    CXOR        BR
               CLOG 116   *
               CLOG 117   *           OR    RX
0CE4   B3F6    CLOG 118    RXOP56    CCOM 045  CXGET       BAL
0CE6   6F15    CLOG 119    CXOR                            H1=H1$U1
0CE8   6E05    CLOG 120                                    H0=H0$U0
0CEA   6735    CLOG 121                                    D1=D1$V1
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
0CEC    6625    CLOG 122                                     DO=DO$VO
0CEE    A844    CLOG 123                  087   CCLOST       BR
                CLOG 124    *
                CLOG 125    *           OR     RS
2B72    AB25    CLOG 126     RSOP96     CICY 056  ADDERR N  BR IF H1=NZ
2B74    5F10    CLOG 127                                     RDB H1 U
2B76    6F55    CLOG 128                                     H1=H1$G1
2B78    A852    CLOG 129                  096   OP94A        BR
                CLOG 130    *
                CLOG 131    *           OR     SS
2988    6F75    CLOG 132     SSOPD6                          H1=H1$D1
298A    AB25    CLOG 133                CICY 056  ADDERR N  BR IF H1=NZ
298C    10CE    CLOG 134                                     RST S K=FC
298E    5738    CLOG 135     OPD6                            RDB D1 V+1
2990    5F10    CLOG 136                                     RDB H1 U
2992    67F5    CLOG 137                                     D1=D1$H1
2994    7718    CLOG 138                                     STB D1 U+1
2996    25FF    CLOG 139                                     G1=G1+KFF
2998    F48F    CLOG 140                  135   OPD6         BR IF AC=1
299A    A848    CLOG 141                  089   S2CHK        BR
                CLOG 142    *
                CLOG 143    *           XOR    RR
348C    8270    CLOG 144     RROP17     CCOM 036  CRGET      BAL                    XOR
348E    8116    CLOG 145                  149   CXXOR        BR                      EXECUTION
                CLOG 146    *
                CLOG 147    *           XOR    RX
0114    B3F6    CLOG 148     RXOP57     CCOM 045  CXGET      BAL
0116    6F11    CLOG 149     CXXOR                           H1=H1<U1
0118    6E01    CLOG 150                                     H0=H0<U0
011A    6731    CLOG 151                                     D1=D1<V1
011C    6621    CLOG 152                                     D0=D0<V0
011E    A844    CLOG 153                  087   CCLOST       BR
                CLOG 154    *
                CLOG 155    *           XOR    RS
2958    AB25    CLOG 156     RSOP97     CICY 056  ADDERR N  BR IF H1=NZ
295A    5F10    CLOG 157                                     RDB H1 U
295C    6F51    CLOG 158                                     H1=H1<G1
295E    A852    CLOG 159                  096   OP94A        BR
                CLOG 160    *
                CLOG 161    *           XOR    SS
2D62    6F75    CLOG 162     SSOPD7                          H1=H1$D1
2D64    AB25    CLOG 163                CICY 056  ADDERR N  BR IF H1=NZ
2D66    10CE    CLOG 164                                     RST S K=FC
2D68    5738    CLOG 165     OPD7                            RDB D1 V+1
2D6A    5F10    CLOG 166                                     RDB H1 U
2D6C    67F1    CLOG 167                                     D1=D1<H1
2D6E    7718    CLOG 168                                     STB D1 U+1
2D70    25FF    CLOG 169                                     G1=G1+KFF
2D72    F4E9    CLOG 170                  165   OPD7         BR IF AC=1
2D74    A848    CLOG 171                  089   S2CHK        BR
                CLOG 172    *
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                CLOG 173    *            TEST UNDER MASK
285A    AB25    CLOG 174     RSOP91    CICY 056  ADDERR N   BR IF H1=NZ
285C    5F10    CLOG 175                                     RDB H1 U
285E    6F57    CLOG 176                                     H1=H1*G1
2860    C4D7    CLOG 177                  098     OP94B      BR IF Z=0
2862    6F51    CLOG 178                                     H1=H1<G1
2864    AB21    CLOG 179                  061     OP95C  N   BR IF H1=NZ
2866    2C75    CLOG 180                                     P0=0$K70               SET CC 3
2868    8185    CLOG 181              CICY 005    ISTART N   N=S BITS67
                CLOG 182    *
                CLOG 183    *            INSERT CHARACTER
2874    AB25    CLOG 184     RXOP43    CICY 056  ADDERR N   BR IF H1=NZ
2876    55BB    CLOG 185                                     T1=G1H
2878    2B3B    CLOG 186                                     T1=T1+K03
287A    5710    CLOG 187                                     RDB D1 U
287C    77A0    CLOG 188                                     STB D1 AS,T
287E    8185    CLOG 189              CICY 005    ISTART N   N=S BITS67
                CLOG 190    *
                CLOG 191    *            STORE CHARACTER
2EA6    AB25    CLOG 192     RXOP42    CICY 056  ADDERR N   BR IF H1=NZ
2EA8    55BB    CLOG 193                                     T1=G1H
2EAA    2B3B    CLOG 194                                     T1=T1+K03
2EAC    57A0    CLOG 195                                     RDB D1 AS,T
2EAE    7710    CLOG 196                                     STB D1 U
2EB0    8185    CLOG 197              CICY 005    ISTART N   N=S BITS67
                CLOG 198    *
                CLOG 199    *               LOAD ADDRESS
2D76    55BB    CLOG 200     RXOP41                          T1=G1H
2D78    2E07    CLOG 201                                     H0=0
2D7A    7EA8    CLOG 202                                     STH H AS,T+2
2D7C    70A0    CLOG 203                                     STH U AS,T
2D7E    8185    CLOG 204              CICY 005    ISTART N   N=S BITS67
                                      **********************************
                                      * CROSS REFERENCE FOR CSECT CLOG *
                                      **********************************

CLOG 004    CICY 227
CLOG 008    CICY 253
CLOG 010    CLOG 013
CLOG 016    CICY 252
CLOG 018    CLOG 023
CLOG 026    CICY 254
CLOG 028    CLOG 033
CLOG 047    CICY 174
CLOG 050    CICY 206
CLOG 051    CLOG 048
CLOG 056    CICY 230
CLOG 060    CLOG 072
CLOG 061    CFAD 079  CLOG 054  CLOG 089  CLOG 097  CLOG 179  CSAS 103  CSFT 076  CTRT 032  CTRT 062
CLOG 063    CFAD 078  CLOG 053  CLOG 060  CSAS 102  CSFT 085  CTRT 031  CTRT 063
CLOG 067    CICY 256
CLOG 069    CLOG 074
```

****
* CROSS REFERENCE FOR CSECT CLOG *
****

```
CLOG 078     CICY 173
CLOG 082     CICY 205
CLOG 083     CLOG 079
CLOG 087     CLOG 123  CLOG 153
CLOG 089     CLOG 111  CLOG 141  CLOG 171
CLOG 093     CICY 229
CLOG 096     CLOG 129  CLOG 159
CLOG 098     CLOG 052  CLOG 059  CLOG 075  CLOG 090  CLOG 177
CLOG 102     CICY 255
CLOG 105     CLOG 110
CLOG 114     CICY 175
CLOG 118     CICY 207
CLOG 119     CLOG 115
CLOG 126     CICY 231
CLOG 132     CICY 257
CLOG 135     CLOG 140
CLOG 144     CICY 176
CLOG 148     CICY 208
CLOG 149     CLOG 145
CLOG 156     CICY 232
CLOG 162     CICY 258
CLOG 165     CLOG 170
CLOG 174     CICY 226
CLOG 184     CICY 188
CLOG 192     CICY 187
CLOG 200     CICY 186
```

CLST -- LOAD AND STORE INSTRUCTIONS

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP10	LOAD POSITIVE	10	R1 R2		
RROP11	LOAD NEGATIVE	11	R1 R2		
RROP12	LOAD AND TEST	12	R1 R2		
RROP13	LOAD COMPLEMENT	13	R1 R2		
RROP18	LOAD	18	R1 R2		
RSOP90	STORE MULTIPLE	90	R1 R3	B2 D2	D2
RSOP98	LOAD MULTIPLE	98	R1 R3	B2 D2	D2
RXOP40	STORE HALFWORD	40	R1 X2	B2 D2	D2
RXOP48	LOAD HALFWORD	48	R1 X2	B2 D2	D2
RXOP50	STORE	50	R1 X2	B2 D2	D2
RXOP58	LOAD	58	R1 X2	B2 D2	D2

OBJECTIVES

LOAD POSITIVE -- THE TRUE VALUE OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.

LOAD NEGATIVE - THE TWOS COMPLIMENT OF THE TRUE VALUE OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.

LOAD COMPLEMENT - THE TWOS COMPLEMENT OF THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION.

LOAD - PLACE THE 2ND OPERAND INTO THE 1ST OPERAND LOCATION.

OBJECTIVES CONTINUED

LOAD AND TEST - THE 2ND OPERAND IS PLACED IN THE 1ST OPERAND LOCATION, AND THE SIGN AND THE MAGNITUDE OF THE 2ND OPERAND DETERMINE THE CONDITION CODE.

LOAD HALFWORD - PLACE THE HALFWORD 2ND OPERAND INTO THE 1ST OPERAND LOCATION. EXTEND THE SIGN INTO THE HIGH-ORDER 16 BITS OF THE 1ST OPERAND.

LOAD MULTIPLE - THE INFORMATION AT THE EFFECTIVE 2ND OPERAND ADDRESS IS LOADED INTO GENERAL REGISTER STARTING WITH THE ONE SPECIFIED BY R1 AND ENDING WITH THE ONE SPECIFIED BY R3.

STORE MULTIPLE - THE INFORMATION IN THE GENERAL REGISTER STARTING WITH THE REGISTER SPECIFIED BY R1 AND AND ENDING WITH THE REGISTER SPECIFIED BY R3 IS STORED AT THE LOCATIONS DESIGNATED BY THE EFFECTIVE 2ND OPERAND ADDRESS.

STORE HALFWORD - MOVE THE 16 LOW-ORDER BYTES OF THE 1ST OPERAND TO PROGRAM STORAGE ADDRESSED BY THE 2ND OPERAND.

STORE - THE 1ST OPERAND IS MOVED TO PROGRAM STORAGE AT THE EFFECTIVE ADDRESS OF THE 2ND OPERAND.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

                CLST 001   T              LOAD AND STORE INSTRUCTIONS
                CLST 002   *
                CLST 003   *
                CLST 004   *            LOAD POSITIVE
2C34   8AC8     CLST 005    RROP10    CCOM 029  LGET       BAL
2C36   C610     CLST 006                   020  DCHECK     BR IF DCO=0
2C38   B3DA     CLST 007    PCOMPL    CCOM 076  COMPL      BAL
2C3A   C610     CLST 008                   020  DCHECK     BR IF DCO=0
2C3C   76A8     CLST 009    OVCHEK                          STH D AS,T+2
2C3E   7EAA     CLST 010                                    STH H AS,T-2
2C40   2C75     CLST 011    SHFTOV                          PO=0$K70           SET CC=3-OVERFLOW
2C42   CD09     CLST 012                   014  TAKEOV     BR IF P10=1
2C44   8185     CLST 013              CICY 005  ISTART N   N=S BITS67
2C08   2B83     CLST 014    TAKEOV                          T1=0$K08
2C0A   89FC     CLST 015              BPSW 003  PROG       BR
                CLST 016   *
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CLST 017   *              LOAD AND TEST
2C0C    8AC8  CLST 018    RROP12   CCOM 029  LGET       BAL
2C0E    C625  CLST 019    DBRAN         036  NEGCC1     BR IF D00=1
2C10    4EE6  CLST 020    DCHECK                        H=H
2C12    4666  CLST 021                                  D=D
2C14    D5C7  CLST 022                 024  TESCC2     BR IF S1=1
2C16    AC30  CLST 023                 034  NEGCC0     BR
2C46    2C25  CLST 024    TESCC2                        P0=0$K20
2C48    76A8  CLST 025    DSTORE                        STH D AS,T+2
2C4A    7EAA  CLST 026                                  STH H AS,T-2
2C4C    8185  CLST 027              CICY 005  ISTART N  N=S BITS67
              CLST 028   *
              CLST 029   *              LOAD NEGATIVE
2C28    8AC8  CLST 030    RROP11   CCOM 029  LGET       BAL
2C2A    C625  CLST 031                 036  NEGCC1     BR IF D00=1
2C2C    B3DA  CLST 032           CCOM 076  COMPL      BAL
2C2E    C625  CLST 033                 036  NEGCC1     BR IF D00=1
2C30    2C07  CLST 034    NEGCC0                        P0=0
2C32    AC48  CLST 035                 025  DSTORE     BR
2C24    2C55  CLST 036    NEGCC1                        P0=0$K50
2C26    AC48  CLST 037                 025  DSTORE     BR
              CLST 038   *
              CLST 039   *              LOAD COMPLIMENT
2C00    8AC8  CLST 040    RROP13   CCOM 029  LGET       BAL
2C02    C639  CLST 041                 007  PCOMPL     BR IF D00=1
2C04    B3DA  CLST 042           CCOM 076  COMPL      BAL
2C06    AC0E  CLST 043                 019  DBRAN      BR
              CLST 044   *
              CLST 045   *              LOAD OP  RR
2E98    55B3  CLST 046    RROP18                        T1=G1XH
2E9A    56A8  CLST 047                                  RDH D AS,T+2
2E9C    5EAA  CLST 048                                  RDH H AS,T-2
2E9E    55BB  CLST 049                                  T1=G1H
2EA0    76A8  CLST 050                                  STH D AS,T+2
2EA2    7EAA  CLST 051                                  STH H AS,T-2
2EA4    8185  CLST 052              CICY 005  ISTART N  N=S BITS67
              CLST 053   *
              CLST 054   *              STORE HALFWORD
2CD4    AB25  CLST 055    RXOP40   CICY 056  ADDERR N   BR IF H1=NZ
2CD6    01E3  CLST 056                                  Z=U1*-K0E
2CD8    F0E4  CLST 057                 063  SPEC       BR IF LZNZ
2CDA    55BB  CLST 058                                  T1=G1H
2CDC    2B2B  CLST 059                                  T1=T1+K02
2CDE    52A0  CLST 060                                  RDH V AS,T+0
2CE0    7210  CLST 061                                  STH V U+0
2CE2    8185  CLST 062    RSEND    CICY 005  ISTART N  N=S BITS67
2CE4    AE94  CLST 063    SPEC     CICY 070  SPECHK     BR
              CLST 064   *
              CLST 065   *              LOAD HALFWORD
2CC0    AB25  CLST 066    RXOP48   CICY 056  ADDERR N   BR IF H1=NZ
2CC2    01E3  CLST 067                                  Z=U1*-K0E
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

2CC4    F0E4    CLST 068                 063     SPEC       BR IF LZNZ
2CC6    55BB    CLST 069                                    T1=G1H
2CC8    5E10    CLST 070                                    RDH H U
2CCA    2707    CLST 071                                    D1=0
2CCC    CE50    CLST 072                 074     EXT        BR IF H00=0
2CCE    27F7    CLST 073                                    D1=0$KFF
2CD0    5769    CLST 074    EXT                             D0=D1
2CD2    AC48    CLST 075                 025     DSTORE     BR
                CLST 076    *
                CLST 077    *          STORE  OP  RX
2EBE    55BB    CLST 078    RXOP50                          T1=G1H
2EC0    56A8    CLST 079                                    RDH D AS,T+2
2EC2    5EAA    CLST 080                                    RDH H AS,T-2
2EC4    7618    CLST 081                                    STH D U+2
2EC6    7E10    CLST 082                                    STH H U
2EC8    8185    CLST 083             CICY 005   ISTART N    N=S BITS67
                CLST 084    *
                CLST 085    *          LOAD OP    RX
2EB2    55BB    CLST 086    RXOP58                          T1=G1H
2EB4    5618    CLST 087                                    RDH D U+2
2EB6    5E10    CLST 088                                    RDH H U
2EB8    76A8    CLST 089                                    STH D AS,T+2
2EBA    7EA0    CLST 090                                    STH H AS,T
2EBC    8185    CLST 091             CICY 005   ISTART N    N=S BITS67
                CLST 092    *
                CLST 093    *          STORE MULTIPLE
3412    ACE6    CLST 094    RSOP90      110     RS9XA      BAL
3414    52A8    CLST 095    RS90A                           RDH V AS,T+2
3416    7218    CLST 096                                    STH V U+2
3418    52A0    CLST 097                                    RDH V AS,T
341A    7218    CLST 098                                    STH V U+2
341C    ACF0    CLST 099                 115     RS9XB      BAL
341E    B414    CLST 100                 095     RS90A      BR
                CLST 101    *
                CLST 102    *          LOAD MULTIPLE
3420    ACE6    CLST 103    RSOP98      110     RS9XA      BAL
3422    5218    CLST 104    RS98A                           RDH V U+2
3424    72A8    CLST 105                                    STH V AS,T+2
3426    5218    CLST 106                                    RDH V U+2
3428    72A0    CLST 107                                    STH V AS,T
342A    ACF0    CLST 108                 115     RS9XB      BAL
342C    B422    CLST 109                 104     RS98A      BR
2CE6    AB25    CLST 110    RS9XA    CICY 056   ADDERR N   BR IF H1=NZ
2CE8    01C3    CLST 111                                    Z=U1*-K0C
2CEA    F0E4    CLST 112                 063     SPEC       BR IF LZNZ
2CEC    55BB    CLST 113                                    T1=G1H
2CEE    128E    CLST 114                                    RTN
2CF0    5531    CLST 115    RS9XB                           V1=G1X
2CF2    6351    CLST 116                                    V1=V1<G1
2CF4    C4E3    CLST 117                 062     RSEND      BR IF Z=0
2CF6    251D    CLST 118                                    G1=G1+K10
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

2CF8    55BB     CLST 119                                    T1=G1H
2CFA    128E     CLST 120                                    RTN
                                      ************************************
                                      * CROSS REFERENCE FOR CSECT CLST *
                                      ************************************

CLST 005     CICY 169
CLST 007     CLST 041
CLST 009     CBIN 028
CLST 011     CSET 074
CLST 014     CLST 012
CLST 018     CICY 171
CLST 019     CLST 043
CLST 020     CLST 006  CLST 008
CLST 024     CBIN 031  CLST 022
CLST 025     CLST 035  CLST 037  CLST 075
CLST 030     CICY 170
CLST 034     CBIN 029  CLST 023
CLST 036     CBIN 030  CLST 019  CLST 031  CLST 033
CLST 040     CICY 172
CLST 046     CICY 177
CLST 055     CICY 185
CLST 062     CLST 117
CLST 063     CLST 057  CLST 068  CLST 112
CLST 066     CICY 193
CLST 074     CLST 072
CLST 078     CICY 201
CLST 086     CICY 209
CLST 094     CICY 225
CLST 095     CLST 100
CLST 103     CICY 233
CLST 104     CLST 109
CLST 110     CLST 094  CLST 103
CLST 115     CLST 099  CLST 108
```

CMLT -- MULTIPLY BINARY ROUTINE

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RROP1C	MULTIPLY	1C	R1 R2		
RXOP4C	MULTIPLY HALFWORD.	4C	R1 X2	B2 D2	D2
RXOP5C	MULTIPLY FULLWORD.	5C	R1 X2	B2 D2	D2

OBJECTIVES

1. FETCH BOTH OPERANDS. FOR FULLWORD MULTIPLY, THE MULTIPLICAND IS FOUND IN THE ODD REGISTER FOLLOWING THE EVEN REGISTER SPECIFIED BY R1. FOR MULTIPLY HALFWORD, R1 SPECIFIES THE LOCATION OF THE MULTIPLICAND. THE MULTIPLIER IS FOUND IN R2 OR AT THE EFFECTIVE MAIN STORAGE ADDRESS.

2. CREATE A TABLE OF MULTIPLICAND VALUES IN AUX STORAGE -- MULTIPLICAND TIMES 1, X4, X16, AND X64.

3. READ EACH HEXADECIMAL MULTIPLIER DIGIT, STARTING AT THE LOW-ORDER. CREATE PRODUCT BY ADDING APPROPRIATE TABLE VALUES OR COMBINATION OF VALUES AS DETERMINED BY THE MULTIPLIER DIGIT. THE PRODUCT IS SHIFTED AFTER ADDING THE TABLE VALUES FOR BOTH THE HIGH AND LOW DIGITS OF EACH MULTIPLIER BYTE.

4. STORE THE PRODUCT IN THE LOCATION THAT ORIGINALLY CONTAINED THE MULTIPLICAND.

DESCRIPTION

THE ROUTINE FETCHES THE OPERANDS, THEN PERFORMS SIGN ANALYSIS. THIS RESULT IS STORED FOR USE AT THE END OF THE ROUTINE.

THE MULTIPLY TABLE IS STORED IN AUX STORAGE. TO BUILD THE TABLE, THE ROUTINE MAKES 4 PASSES THROUGH A LOOP. IN THE LOOP, THE MULTIPLICAND IS STORED, THEN DOUBLED AND REDOUBLED. THEREFORE, THE VALUES STORED IN THE TABLE ARE -- MULTIPLICAND TIMES 1, 4, 16, AND 64. SEE AUX STORAGE MAP IN SECTION AAAB FOR LOCATION.

MULTIPLICATION IS DONE ON A BYTE BASIS. FOR THE LOW DIGIT OF THE MULTIPLIER BYTE, THE TABLE ADDRESS IS INITIALIZED FOR X1. FOR THE HIGH DIGIT, THE TABLE ADDRESS IS SET FOR X16. THEN, A 16-WAY BRANCH ON THE ACTUAL MULTIPLIER DIGIT OCCURS. THE BRANCH DIRECTS THE MICROROUTINE TO THE PROPER COMPUTE ROUTINE FOR THE CURRENT MULTIPLIER DIGIT. THE COMPUTE ROUTINES CHANGE THE MULTIPLY-TABLE ADDRESS IF A X4 OR A X64 TABLE VALUE IS NEEDED. ALSO, COMPLEMENT ADD IS SET IF A SUBTRACT IS NEEDED. THEN THE MICROROUTINE IS DIRECTED TO ADD EITHER ONCE OR TWICE DEPENDING ON THE COMPUTE ROUTINE FOR THE CURRENT DIGIT. NOTE THAT FOR SOME DIGITS, ADDS FROM TWO DIFFERENT TABLE ADDRESSES ARE NECESSARY.

SHIFTING OCCURS AFTER BOTH THE HIGH AND LOW DIGITS FOR EACH MULTIPLIER BYTE ARE PROCESSED. THE CURRENTLY COMPLETED PRODUCT BYTE IS SHIFTED OUT OF THE PRODUCT-BUILDING AREA AND INTO THE REGISTERS FORMERLY OCCUPIED BY THE MULTIPLIER.

A COUNTER DETERMINES THE LAST MULTIPLIER DIGIT. AFTER THIS LAST DIGIT IS PROCESSED, A CARRY TEST IS MADE. A FINAL X1 VALUE FROM THE TABLE IS COMPLEMENT-ADDED TO THE PRODUCT IF THE NEED IS INDICATED BY THE SIGN AND MULTIPLIER-LOW DIGIT CARRY.

TO COMPLETE THE MULTIPLY OPERATION, THE MICROPROGRAM RESTORES THE SAVED REGISTERS, AND THE HALFWORD OR FULLWORD PRODUCT IS STORED AT THE 1ST OPERAND ADDRESS. THEN, THE MICROPROGRAM RETURNS TO I-CYCLES.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS

              CMLT 001     T                    MULTIPLY BINARY OP CODES        D. L. SMITH
2064    F563  CMLT 002      RXOP5C         006   ERRORA     BR IF G13=1
2066    3515  CMLT 003                                      G1=G1$K10
2068    B3F6  CMLT 004                 CCOM 045   CXGET      BAL
206A    897A  CMLT 005                      015   START      BR
2062    AE94  CMLT 006      ERRORA    CICY 070   SPECHK     BR
206C    E1F1  CMLT 007      RROP1C         009   NEQLAB     BR IF S6=1
206E    3485  CMLT 008                                      G0=G0$K80
2070    F563  CMLT 009      NEQLAB         006   ERRORA     BR IF G13=1
2072    3515  CMLT 010                                      G1=G1$K10
2074    8270  CMLT 011                 CCOM 036   CRGET      BAL
2076    897A  CMLT 012                      015   START      BR
              CMLT 013     ASEQ    ALO7=78
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0978   AE06    CMLT 014    RXOP4C   CCOM 012  CXGETH     BAL
097A   9814    CMLT 015    START    CCOM 057  LSAVEA     BAL
097C   2907    CMLT 016                                   I1=0
097E   C604    CMLT 017                 020   CANDPL     BR IF D00=0
0980   29F7    CMLT 018                                   I1=0$KFF
0982   2C13    CMLT 019                                   P0=0$K01
0984   0275    CMLT 020    CANDPL                         Z=V0*-K70
0986   E08B    CMLT 021                 023   PLIERP     BR IF HZ=0
0988   3C23    CMLT 022                                   P0=P0$K02
098A   5989    CMLT 023    PLIERP                         I0=I1
098C   4A86    CMLT 024                                   T=I
098E   00C0    CMLT 025                                   RST S K=0C
0990   7CF2    CMLT 026    LOOP1                          STH P DA,BE
0992   4C52    CMLT 027    FLTMU1                         RDH P DC,9A
0994   2D1D    CMLT 028    LOOP                           P1=P1+K10
0996   2D8B    CMLT 029                                   P1=P1+K08
0998   CD29    CMLT 030                 038   EXIT       BR IF P10=1
099A   78C8    CMLT 031                                   STH I AS,P+2
099C   7AC8    CMLT 032                                   STH T AS,P+2
099E   76C8    CMLT 033                                   STH D AS,P+2
09A0   7EC8    CMLT 034                                   STH H AS,P+2
09A2   9AFA    CMLT 035                 104   DOUBLE     BAL
09A4   9AFA    CMLT 036                 104   DOUBLE     BAL
09A6   8994    CMLT 037                 028   LOOP       BR
09A8   2D07    CMLT 038    EXIT                           P1=0
09AA   48C6    CMLT 039                                   I=P
09AC   4AC6    CMLT 040                                   T=P
09AE   44C6    CMLT 041                                   G=P
09B0   46C6    CMLT 042                                   D=P
09B2   4E06    CMLT 043                                   H=U
09B4   40C6    CMLT 044                                   U=P
09B6   2C13    CMLT 045                                   P0=0$K01
09B8   C5BE    CMLT 046    CHKLO4        049   LOCHK      BR IF S0=0
09BA   2F1B    CMLT 047    NXY                            H1=H1+K01
09BC   F4D3    CMLT 048                 059   ADJUST     BR IF AC=1
09BE   2D15    CMLT 049    LOCHK                          P1=0$K10
09C0   1002    CMLT 050                                   RST S K=90
09C2   BB03    CMLT 051                 134   DIG     N  N=H1L
09C4   DD53    CMLT 052    CYCLE        059   ADJUST     BR IF P11=1
09C6   C5CC    CMLT 053                 056   HICHK      BR IF S0=0
09C8   2F1D    CMLT 054                                   H1=H1+K10
09CA   F4D3    CMLT 055                 059   ADJUST     BR IF AC=1
09CC   2D55    CMLT 056    HICHK                          P1=0$K50
09CE   1002    CMLT 057                                   RST S K=90
09D0   BF03    CMLT 058                 134   DIG     N  N=H1H
09D2   2C2B    CMLT 059    ADJUST                         P0=P0+K02
09D4   AF4A    CMLT 060                 085   ROTA       BAL
09D6   C838    CMLT 061    FLTMU2       046   CHKLO4     BR IF P04=0
09D8   C1DC    CMLT 063                 065   ALMOST     BR IF S4=0
09DA   A354    CMLT 064             CFMU 009  MULT       BR                 EXIT TO FLOAT PT MULT
09DC   54F2    CMLT 065    ALMOST                         RDH G DA,BE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS
09DE    C5E9    CMLT 069                  074   ANSWER     BR IF S0=1
09E0    E068    CMLT 070                  074   ANSWER     BR IF G06=0
09E2    3002    CMLT 071                                   SET S K=90
09E4    2D15    CMLT 072                                   P1=0$K10
09E6    A752    CMLT 073                  113   ADD1       BAL
09E8    8176    CMLT 074    ANSWER   CCOM 065   LRSTRA     BAL
09EA    2AC7    CMLT 075                                   T0=0
09EC    55BB    CMLT 076                                   T1=G1H
09EE    72A8    CMLT 077                                   STH V AS,T+2
09F0    7EAA    CMLT 078                                   STH H AS,T-2
09F2    F47A    CMLT 079                  083   DONE       BR IF G03=0
09F4    1B15    CMLT 080                                   T1=T1*-K10
09F6    76A8    CMLT 081                                   STH D AS,T+2
09F8    70AA    CMLT 082                                   STH U AS,T-2
09FA    AA76    CMLT 083    DONE     CICY 004   CHECK      BR
                CMLT 084   AEND
2F4A    5EF9    CMLT 085    ROTA                           H1=H0
2F4C    53E9    CMLT 086                                   H0=V1
2F4E    5239    CMLT 087                                   V1=V0
2F50    5129    CMLT 088                                   V0=U1
2F52    5019    CMLT 089                                   U1=U0
2F54    5709    CMLT 090                                   U0=D1
2F56    5679    CMLT 091                                   D1=D0
2F58    5B69    CMLT 092                                   D0=T1
2F5A    5AB9    CMLT 093                                   T1=T0
2F5C    55A9    CMLT 094                                   T0=G1
2F5E    5459    CMLT 095                                   G1=G0
2F60    2407    CMLT 096                                   G0=0
2F62    C56E    CMLT 097                  103   NOFLOW     BR IF G10=0
2F64    058F    CMLT 098                                   Z=G1<K88
2F66    E0EC    CMLT 099                  102   FLOFIX     BR IF HZNZ
2F68    C16E    CMLT 100                  103   NOFLOW     BR IF G14=0
2F6A    C5EE    CMLT 101                  103   NOFLOW     BR IF S0=0
2F6C    24F7    CMLT 102    FLOFIX                         G0=0$KFF
2F6E    128E    CMLT 103    NOFLOW                         RTN
1AFA    6FFB    CMLT 104    DOUBLE                         H1C=H1+H1
1AFC    6EED    CMLT 105                                   H0C=H0+H0+C
1AFE    677D    CMLT 106                                   D1C=D1+D1+C
1B00    666D    CMLT 107                                   D0C=D0+D0+C
1B02    6BBD    CMLT 108                                   T1C=T1+T1+C
1B04    6AAD    CMLT 109                                   T0C=T0+T0+C
1B06    699D    CMLT 110                                   I1C=I1+I1+C
1B08    688D    CMLT 111                                   I0C=I0+I0+C
1B0A    128E    CMLT 112                                   RTN
2752    1CFB    CMLT 113    ADD1                           P0=P0<K0F
2754    3DE3    CMLT 114    ADDROU                         P1=P1$K0E
2756    58CA    CMLT 115                                   RDH I AS,P-2
2758    719D    CMLT 116                                   U1C=U1%I1+C
275A    708D    CMLT 117                                   U0C=U0%I0+C
275C    58CA    CMLT 118                                   RDH I AS,P-2
275E    779D    CMLT 119                                   D1C=D1%I1+C
```

ADDR	WORD	SEQUENCE NO.	LABEL		NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
2760	768D	CMLT 120					DOC=DO%IO+C	
2762	58CA	CMLT 121					RDH I AS,P-2	
2764	7B9D	CMLT 122					T1C=T1%I1+C	
2766	7A8D	CMLT 123					TOC=TO%IO+C	
2768	58CA	CMLT 124					RDH I AS,P-2	
276A	759D	CMLT 125					G1C=G1%I1+C	
276C	748D	CMLT 126					GOC=GO%IO+C	
276E	1CFB	CMLT 127					PO=PC<KOF	
2770	0002	CMLT 128					RST S3	
2772	F87B	CMLT 129			133	OUTOUT	BR IF PO7=1	
2774	C5D4	CMLT 130			114	ADDROU	BR IF SO=0	
2776	2002	CMLT 131					SET S3	
2778	C5D5	CMLT 132			114	ADDROU	BR IF SO=1	
277A	128E	CMLT 133	OUTOUT				RTN	
0900	89C4	CMLT 134	DIG	0	052	CYCLE	BR	0
0902	8920	CMLT 135	DIG	1	150	PUTE1	BR	+16 OR 1
0904	A37A	CMLT 136	DIG	2	167	PUTE2	BR	&32 OR 2
0906	B46C	CMLT 137	DIG	3	152	PUTE3	BR	+48 OR 3
0908	B472	CMLT 138	DIG	4	155	PUTE4	BR	+64 OR 4
090A	B46E	CMLT 139	DIG	5	153	PUTE5	BR	+80 OR 5
090C	A200	CMLT 140	DIG	6	158	PUTE6	BR	+96 OR 6
090E	A372	CMLT 141	DIG	7	163	PUTE7	BR	+112 OR 7
0910	A378	CMLT 142	DIG	8	166	PUTE8	BR	+128 OR 8
0912	903A	CMLT 143	DIG	9	169	PUTE9	BR	-112 OR 7
0914	8212	CMLT 144	DIG	A	172	PUTEA	BR	-96 OR 6
0916	83A0	CMLT 145	DIG	B	181	PUTEB	BR	-80 OR 5
0918	AFFC	CMLT 146	DIG	C	179	PUTEC	BR	-64 OR 4
091A	83A2	CMLT 147	DIG	D	182	PUTED	BR	-48 OR 3
091C	821A	CMLT 148	DIG	E	176	PUTEE	BR	-32 OR 2
091E	3002	CMLT 149	DIG	F			SET S K=90	-16 OR 1
0920	A752	CMLT 150	PUTE1		113	ADD1	BAL	
0922	89C4	CMLT 151			052	CYCLE	BR	
346C	3002	CMLT 152	PUTE3				SET S K=90	
346E	A752	CMLT 153	PUTE5		113	ADD1	BAL	
3470	1002	CMLT 154					RST S K=90	
3472	3D25	CMLT 155	PUTE4				P1=P1$K20	
3474	A752	CMLT 156			113	ADD1	BAL	
3476	89C4	CMLT 157			052	CYCLE	BR	
2200	3D25	CMLT 158	PUTE6				P1=P1$K20	
2202	A752	CMLT 159			113	ADD1	BAL	
2204	1D25	CMLT 160					P1=P1*-K20	
2206	A754	CMLT 161			114	ADDROU	BAL	
2208	89C4	CMLT 162			052	CYCLE	BR	
2372	3002	CMLT 163	PUTE7				SET S K=90	
2374	A752	CMLT 164			113	ADD1	BAL	
2376	1002	CMLT 165					RST S K=90	
2378	3D25	CMLT 166	PUTE8				P1=P1$K20	
237A	A754	CMLT 167	PUTE2		114	ADDROU	BAL	
237C	89C4	CMLT 168			052	CYCLE	BR	
103A	A752	CMLT 169	PUTE9		113	ADD1	BAL	
103C	3002	CMLT 170					SET S K=90	

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

103E    A378    CMLT 171                  166   PUTE8      BR
0212    3002    CMLT 172    PUTEA                          SET S K=90
0214    3D25    CMLT 173                                   P1=P1$K20
0216    A752    CMLT 174                  113   ADD1       BAL
0218    1D25    CMLT 175                                   P1=P1*-K20
021A    3002    CMLT 176    PUTEE                          SET S K=90
021C    A754    CMLT 177                  114   ADDROU     BAL
021E    89C4    CMLT 178                  052   CYCLE      BR
2FFC    3002    CMLT 179    PUTEC                          SET S K=90
2FFE    B472    CMLT 180                  155   PUTE4      BR
03A0    3002    CMLT 181    PUTEB                          SET S K=90
03A2    A752    CMLT 182    PUTED         113   ADD1       BAL
03A4    3002    CMLT 183                                   SET S K=90
03A6    B472    CMLT 184                  155   PUTE4      BR
                                          ************************************
                                          * CROSS REFERENCE FOR CSECT CMLT *
                                          ************************************

CMLT 002    CICY 213
CMLT 006    CMLT 002  CMLT 009
CMLT 007    CICY 181
CMLT 009    CMLT 007
CMLT 014    CICY 197
CMLT 015    CMLT 005  CMLT 012
CMLT 020    CMLT 017
CMLT 023    CMLT 021
CMLT 027    CFMD 065  CFMD 075
CMLT 028    CMLT 037
CMLT 038    CMLT 030
CMLT 046    CMLT 061
CMLT 049    CMLT 046
CMLT 052    CMLT 134  CMLT 151  CMLT 157  CMLT 162  CMLT 168  CMLT 178
CMLT 056    CMLT 053
CMLT 059    CMLT 048  CMLT 052  CMLT 055
CMLT 061    CFMU 032
CMLT 065    CMLT 063
CMLT 074    CMLT 069  CMLT 070
CMLT 083    CMLT 079
CMLT 085    CMLT 060
CMLT 102    CMLT 099
CMLT 103    CMLT 097  CMLT 100  CMLT 101
CMLT 104    CMLT 035  CMLT 036
CMLT 113    CMLT 073  CMLT 150  CMLT 153  CMLT 156  CMLT 159  CMLT 164  CMLT 169  CMLT 174  CMLT 182
CMLT 114    CMLT 130  CMLT 132  CMLT 161  CMLT 167  CMLT 177
CMLT 133    CMLT 129
CMLT 134    CMLT 051  CMLT 058
CMLT 150    CMLT 135
CMLT 152    CMLT 137
CMLT 153    CMLT 139
CMLT 155    CMLT 138  CMLT 180  CMLT 184
CMLT 158    CMLT 140
CMLT 163    CMLT 141
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT CMLT *
                                  **********************************

CMLT  166     CMLT  142  CMLT 171
CMLT  167     CMLT  136
CMLT  169     CMLT  143
CMLT  172     CMLT  144
CMLT  176     CMLT  148
CMLT  179     CMLT  146
CMLT  181     CMLT  145
CMLT  182     CMLT  147
```

CMPU DESCRIPTIVE TEXT

LABEL..SSOPF1 MOVE WITH OFFSET

STEP

1..READ LOW SOURCE BYTE
2..MOVE LOW 4 BITS OF SOURCE BYTE TO HIGH 4 BITS OF THE LOW DESTINATION BYTE. THE LOW 4 BITS OF THE DESTINATION BYTE REMAIN UNCHANGED.
3..CHECK TO SEE WHAT FIELD HAS ENDED,

A. IF DESTINATION OR BOTH ENDED, RETURN TO I-CYCLES.

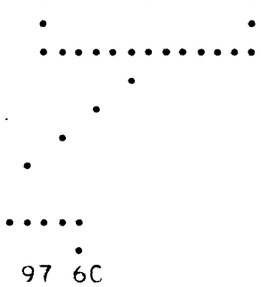
B. IF SOURCE HAS ENDED, STORE THE HIGH 4 BITS OF THE SOURCE BYTE IN THE LOW 4 BITS OF THE NEXT DESTINATION BYTE. FILL DESTINATION REMAINDER WITH ZEROES, RETURN TO I-CYCLES.

C. IF NEITHER HAS ENDED, READOUT NEXT SOURCE BYTE, STORE THE HIGH 4 BITS OF PREVIOUS SOURCE BYTE AS LOW 4 BITS IN DESTINATION BYTE. STORE LOW 4 BITS OF NEW SOURCE BYTE AS HIGH 4 BITS IN DESTINATION BYTE. RETURN TO STEP 3.

MOVE WITH OFFSET EXAMPLE

DESTINATION BEFORE 79 56 87 84 3C

SOURCE 64 72 63 75 91 85 29 76



DESTINATION AFTER 59 18 52 97 6C

LABEL..SSOPF2 PACK

STEP

1..READOUT LOW SOURCE BYTE. CROSS DIGITS AND STORE AT LOW DESTINATION ADDRESS.

2..CHECK TO SEE WHAT FIELD HAS ENDED.

A. IF DESTINATION OR BOTH ENDED, RETURN TO I-CYCLES.

B. IF SOURCE HAS ENDED, FILL REMAINING DESTINATION WITH ZEROES.

C. IF NEITHER HAS ENDED, READOUT NEXT SOURCE BYTE. IF THIS BYTE ENDS SOURCE FIELD, FILL REMAINING DESTINATION WITH ZEROES, AFTER STORING LOW 4 SOURCE BITS. IF SOURCE DID NOT END, READOUT NEXT SOURCE BYTE, COMBINE LOW ORDER SOURCE DIGITS AND STORE AT NEXT HIGHER DESTINATION ADDRESS. RETURN TO STEP 2.

PACK EXAMPLE

SOURCE F6 F7 F2 F4 F9 C3

DESTINATION BEFORE 00 00 00 00 00

DESTINATION AFTER 00 06 72 49 3C

LABEL..SSOPF3 UNPACK

THE UNPACK INSTRUCTION IS ALMOST THE REVERSE OF PACK. PROPER ZONES ARE INSERTED IN DESTINATION BYTES FOR ALL SOURCE DIGITS. ZONES ARE INSERTED WITH ZEROES WHEN FILLING IN DESTINATION FIELD WHEN SOURCE HAS ENDED.

UNPACK EXAMPLE

SOURCE 64 92 5C

DESTINATION BEFORE 00 00 00 00 00 00

DESTINATION AFTER F0 F6 F4 F9 F2 C5

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT            COMMENTS

              CMPU 001    T                        PACK,UNPACK, AND MOVE WITH OFFSET   12/15/66   R. C. HUANG
              CMPU 002    *
0136    583A  CMPU 003     SSOPF1                             RDB T1 V-1           MOVE OFFSET, READ SOURCE SIGN
0138    5F10  CMPU 004                                        RDB H1 U+0           READ DESTINATION SIGN BYTE
013A    48F3  CMPU 005                                        H1=T1XH+H1L          SET UP THE SIGN BYTE
013C    5B75  CMPU 006                                        D1=T1XL              SAVE HI HALF BYTE OF SOURCE
013E    AE72  CMPU 007                    013  STBYTE       BR
2E68    27F5  CMPU 008     SSOPF3                             D1=0$KF0             UNPACK, SET UP FOR EBCDIC MODE
2E6A    C96E  CMPU 009                    011  SSOPF2       BR IF P1 BIT4=0      BR ON NOT ASCII MODE
2E6C    2755  CMPU 010                                        D1=0$K50             SET UP FOR ASCII MODE
2E6E    583A  CMPU 011     SSOPF2                             RDB T1 V-1           PACK, READ SIGN BYTE
2E70    58F1  CMPU 012                                        H1=T1X               CROSS SIGN BYTE
2E72    7F1A  CMPU 013     STBYTE                             STB H1 U-1           STORE SIGN BYTE
2E74    5559  CMPU 014                                        G1=G1                PUT LENGTH BYTE ON Z-BUSS
2E76    8093  CMPU 015                    016  HZLZBR N     N=DYN BITS67         BR ON HZ,LZ - ZERO LENGTH
2980    8033  CMPU 016     HZLZBR 0       020  OP     N     N=GO BITS67          BR ON OP'S
2982    A9B2  CMPU 017     HZLZBR 1       028  STDEST       BR                   SOURCE ENDED
2984    8185  CMPU 018     HZLZBR 2  CICY 005  ISTART N     N=S BITS67           DESTINATION ENDED
2986    8185  CMPU 019     HZLZBR 3  CICY 005  ISTART N     N=S BITS67           BOTH ENDED
29A2    81B6  CMPU 020     OP     1       032  MWOS         BR                   MOVE WITH OFFSET
29A4    AA0E  CMPU 021     OP     2       037  PACK         BR                   PACK
29A6    583A  CMPU 022     OP     3                          RDB T1 V-1           UNPACK, READ SOURCE
29A8    4B7D  CMPU 023                                        D1=T1L+D1H           UNPACK LO HALF BYTE
29AA    771A  CMPU 024                                        STB D1 U-1           STORE FIRST BYTE
29AC    351D  CMPU 025                                        G1=G1-K10            SUBTRACT HI, LO COUNT BY 1
29AE    E087  CMPU 026                    019  HZLZBR 3     BR IF HZ=0           BR IF DESTINATION END
29B0    4B75  CMPU 027                                        D1=T1XL$D1H          UNPACK HI HALF BYTE
29B2    771A  CMPU 028     STDEST                             STB D1 U-1           STORE DESTINATION BYTE
29B4    577B  CMPU 029                                        D1=D1H               ZERO OUT NUMERIC
29B6    25FD  CMPU 030                                        G1=G1+KF0            SUBTRACT DEST COUNT
29B8    8093  CMPU 031                    016  HZLZBR N     N=DYN BITS67         BR ON HZ,LZ COUNT
01B6    583A  CMPU 032     MWOS                               RDB T1 V-1           READ SOURCE
01B8    4B73  CMPU 033                                        D1=T1XH+D1L          SET UP THE BYTE
01BA    771A  CMPU 034                                        STB D1 U-1           STORE DESTINATION
01BC    5B75  CMPU 035                                        D1=T1XL              SAVE HI HALF BYTE FOR NEXT CYCLE
01BE    AA1A  CMPU 036                    043  HILOCT       BR                   GO TO SUBT HI, LO COUNT
2A0E    5F3A  CMPU 037     PACK                               RDB H1 V-1           READ FIRST SOURCE BYTE
2A10    25FF  CMPU 038                                        G1=G1+KFF            SUBTRACT LOW COUNT BY 1
2A12    F08B  CMPU 039                    045  SOUEND       BR IF LZ=0           BR IF SOURCE ENDED
2A14    583A  CMPU 040                                        RDB T1 V-1           READ SECOND SOURCE BYTE
2A16    48F3  CMPU 041                                        H1=T1XH+H1L          PACK THE TWO BYTES
2A18    7F1A  CMPU 042                                        STB H1 U-1           STORE IT
2A1A    351D  CMPU 043     HILOCT                             G1=G1-K10            SUBT HI, LO COUNT BY 1
2A1C    8093  CMPU 044                    016  HZLZBR N     N=DYN BITS67         BR ON HZ, LZ COUNT
2A0A    5F7D  CMPU 045     SOUEND                             D1=H1L               SOURCE ENDED, ZERO OUT HI HALF
2A0C    A9B2  CMPU 046                    028  STDEST       BR                   GO TO STORE & BR HZ
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT CMPU *
                                  **********************************

CMPU 003     CICY 268
CMPU 008     CICY 270
CMPU 011     CICY 269  CMPU 009
CMPU 013     CMPU 007
CMPU 016     CMPU 015  CMPU 031  CMPU 044
CMPU 019     CMPU 026
CMPU 020     CMPU 016
CMPU 028     CMPU 017  CMPU 046
CMPU 032     CMPU 020
CMPU 037     CMPU 021
CMPU 043     CMPU 036
CMPU 045     CMPU 039
```

CNVR DESCRIPTIVE TEXT

ENTRY	OPERATION	INSTRUCTION FORMAT BYTE 1	BYTE 2	BYTE 3	BYTE 4
RXOP4E	CONVERT TO DEC.	4E	R1 X2	B2 D2	D2
RXOP4E	CONVERT TO BIN.	4F	R1 X2	B2 D2	D2

PRIOR TO ENTRY TO THIS ROUTINE, THE EFFECTIVE ADDRESS FOR THE SECOND OPERAND HAS BEEN CALCULATED IN THE CICY ROUTINE.

OBJECTIVES

CONVERT TO BINARY - THE SECOND OPERAND IS CHANGED FROM A PACKED DECIMAL NUMBER TO A BINARY NUMBER AND PLACED IN THE GENERAL PURPOSE REGISTER SPECIFIED BY R1. THE SECOND OPERAND IS CHECKED FOR VALID SIGN AND DIGIT CODES.

CONVERT TO DECIMAL - THE FIRST OPERAND IS CHANGED FROM A SIGNED BINARY NUMBER TO A PACKED DECIMAL NUMBER AND PLACED IN THE SECOND OPERAND LOCATION.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CNVR 001    T              CONVERT TO DECIMAL AND BINARY ROUTINES
              CNVR 002    *
              CNVR 003    *
              CNVR 004    *            CONVERT TO DECIMAL
2C4E    AB25  CNVR 005     RXOP4E    CICY 056  ADDERR N  BR IF H1=NZ
2C50    0183  CNVR 006                                   Z=U1*-K08
2C52    F0D7  CNVR 007                 009  OKAY       BR IF LZ=0
2C54    AE94  CNVR 008               CICY 070  SPECHK     BR
2C56    9814  CNVR 009    OKAY      CCOM 057  LSAVEA     BAL
2C58    10CE  CNVR 010                                   RST S K=FC
2C5A    F05E  CNVR 011                 013  RX4EA      BR IF GC7=0
2C5C    AB7A  CNVR 012                 058  RXOP4F     BR
2C5E    AE16  CNVR 013    RX4EA     CCOM 020  GETR1      BAL
2C60    2807  CNVR 014                                   T1=0                STORE FIRST
2C62    7A18  CNVR 015                                   STH T U+2            TWO ANS. BYTES
2C64    C66A  CNVR 016                 019  RX4EB      BR IF D00=0
2C66    B3DA  CNVR 017               CCOM 076  COMPL      BAL
2C68    2040  CNVR 018                                   SET S5                  SET MINUS
2C6A    42A6  CNVR 019    RX4EB                          V=T
2C6C    44A6  CNVR 020                                   G=T
2C6E    48A6  CNVR 021                                   I=T
2C70    2D13  CNVR 022                                   P1=0$K01
2C72    C679  CNVR 023                 026  L4EB       BR IF D00=1
2C74    AD10  CNVR 024    L4EA         039  DOBBIN     BAL
2C76    C674  CNVR 025                 024  L4EA       BR IF D00=0
2C78    ACFE  CNVR 026    L4EB         030  RX4EAA     BAL
2C7A    C679  CNVR 027    L4EC         026  L4EB       BR IF D00=1
2C7C    AD00  CNVR 028                 031  RX4EAB     BAL
2C7E    AC7A  CNVR 029                 027  L4EC       BR
2CFE    3915  CNVR 030    RX4EAA                         I1=I1$K10
2D00    ED1F  CNVR 031    RX4EAB       046  END        BR IF P12=1
2D02    0002  CNVR 032                                   RST S3
2D04    799F  CNVR 033                                   I1C=I1@I1+C
2D06    788F  CNVR 034                                   I0C=I0@I0+C
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
2D08    755F    CNVR 035                                    G1C=G1@G1+C
2D0A    744F    CNVR 036                                    G0C=G0@G0+C
2D0C    733F    CNVR 037                                    V1C=V1@V1+C
2D0E    722F    CNVR 038                                    V0C=V0@V0+C
2D10    6FFB    CNVR 039    DOBBIN                          H1C=H1+H1
2D12    6EED    CNVR 040                                    H0C=H0+H0+C
2D14    677D    CNVR 041                                    D1C=D1+D1+C
2D16    666D    CNVR 042                                    D0C=D0+D0+C
2D18    ED1F    CNVR 043                 046  END           BR IF P12=1
2D1A    2D1B    CNVR 044                                    P1=P1+K01
2D1C    128E    CNVR 045                                    RTN
2D1E    4686    CNVR 046    END                             D=I
2D20    8178    CNVR 047              CCOM 066  LRSTRB      BAL
2D22    27CB    CNVR 048                                    D1=D1+KCC          SET NOT ASCII PLUS
2D24    C928    CNVR 049                 051  NEXTCK        BR IF P14=0
2D26    176B    CNVR 050                                    D1=D1<K06          SET ASCII PLUS
2D28    D1AC    CNVR 051    NEXTCK       053  FINAL         BR IF S5=0
2D2A    3713    CNVR 052                                    D1=D1$K01          MAKE SIGN MINUS
2D2C    7218    CNVR 053    FINAL                           STH V U+2          STORE
2D2E    7418    CNVR 054                                    STH G U+2           RESULT
2D30    7618    CNVR 055                                    STH D U+2
2D32    8185    CNVR 056              CICY 005  ISTART N    N=S BITS67
                CNVR 057  *               CONVERT TO BINARY
287A    2B83    CNVR 058    RXOP4F                          T1=0$K08           STORE
287C    3815    CNVR 059                                    T1=T1$K10           THE
287E    4EA6    CNVR 060                                    H=T                 DECIMAL
2B80    5718    CNVR 061    LOOPAX                          RDB D1 U+1          FIELD
2B82    77EF    CNVR 062                                    D1C=D1@H0+C         TN
2B84    E589    CNVR 063                 065  NOTSAV        BR IF S2=1           AUX
2B86    5AE4    CNVR 064                                    T=H+1                 STORAGE
2B88    77E8    CNVR 065    NOTSAV                          STB D1 AS,H+1          AND
2B8A    0FFB    CNVR 066                                    Z=H1<K0F               FIND
2B8C    F080    CNVR 067                 061  LOOPAX        BR IF LZNZ              1ST
2B8E    5710    CNVR 068                                    RDB D1 U                 SIG
2B90    573B    CNVR 069                                    V1=D1H                    DIGET
2B92    8F2C    CNVR 070              CDMD 203  SINSET      BAL                GO CHECK SIGN
2B94    73AF    CNVR 071                                    V1C=V1@T0+C
2B96    C19A    CNVR 072                 074  OKFLD         BR IF S4=0
2B98    9BDC    CNVR 073              CCOM 071  FLGDEX       BR                 GO FLAG DATA EXCEP
2B9A    73E0    CNVR 074    OKFLD                           STB V1 AS,H
2B9C    555B    CNVR 075                                    G1=G1H             BUILD
2B9E    3533    CNVR 076                                    G1=G1$K03           ANS
2BA0    10CE    CNVR 077                                    RST S K=FC           REG ADDRESS
2BA2    E426    CNVR 078                 080  PLSNUM        BR IF G02=0        BR IF PLUS
2BA4    3002    CNVR 079                                    SET S K=90
2BA6    2407    CNVR 080    PLSNUM                          G0=0
2BA8    2907    CNVR 081    CONTUE                          I1=0
2BAA    0040    CNVR 082                                    RST S5               INITIALIZE ZERO FLAG
2BAC    4EA6    CNVR 083                                    H=T
2BAE    2207    CNVR 084                                    V0=0
2BB0    5DE0    CNVR 085    RDBP1H                          RDB P1 AS,H
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
2BB2    5DC5    CNVR 086                                   P0=P1XL
2BB4    B070    CNVR 087                 123     X10PLS    BAL
2BB6    527B    CNVR 088                                   D1=V0H
2BB8    0FF8    CNVR 089                                   Z=H1<K0F
2BBA    F0E8    CNVR 090                 100     LAB6      BR IF LZNZ
2BBC    77E0    CNVR 091                                   STB D1 AS,H.
2BBE    14FF    CNVR 092                                   G0=G0<KFF
2BC0    C4CA    CNVR 093                 098     LAB9      BR IF ZNZ
2BC2    4203    CNVR 094                                   U0=V0XH+U0L
2BC4    790D    CNVR 095                                   I1C=I1%U0+C
2BC6    794A    CNVR 096                                   STB I1 AS,G-1
2BC8    C14F    CNVR 097                 110     DONENO    BR IF G14=1
2BCA    520D    CNVR 098    LAB9                           U0=V0L
2BCC    ABA8    CNVR 099                 081     CONTUE    BR
2BE8    5DCD    CNVR 100    LAB6                           P0=P1L
2BEA    B070    CNVR 101                 123     X10PLS    BAL
2BEC    4275    CNVR 102                                   D1=V0XL$D1H
2BEE    D1F5    CNVR 103                 106     LAB7      BR IF S5=1          BR IF NZ RESUT SO FAR
2BF0    C4F8    CNVR 104                 108     LAWOPS    BR IF ZNZ            SET NOT ZERO FLAG
2BF2    5AA4    CNVR 105                                   T=T+1
2BF4    77E8    CNVR 106    LAB7                           STB D1 AS,H+1
2BF6    ABB0    CNVR 107                 085     RDBP1H    BR
2BF8    2040    CNVR 108    LAWOPS                         SET S5
2BFA    ABF4    CNVR 109                 106     LAB7      BR
2BCE    59F9    CNVR 110    DONENO                         H1=I1
2BD0    8176    CNVR 111             CCOM 065    LRSTRA    BAL
2BD2    C5DA    CNVR 112                 116     LAB12     BR IF S0=0
2BD4    E5DC    CNVR 113                 117     LAB13     BR IF S2=0
2BD6    CF62    CNVR 114                 120     LAB14     BR IF H10=0
2BD8    ABDC    CNVR 115                 117     LAB13     BR
2BDA    CF63    CNVR 116    LAB12        120     LAB14     BR IF H10=1
2BDC    47BB    CNVR 117    LAB13                          T1=D1H+T1L
2BDE    0BFB    CNVR 118                                   Z=T1<K0F
2BE0    C4E7    CNVR 119                 122     LAB11     BR IF Z=0
2BE2    2B93    CNVR 120    LAB14                          T1=0$K09
2BE4    89FC    CNVR 121             BPSW 003    PROG      BR
2BE6    8185    CNVR 122    LAB11    CICY 005    ISTART N  N=S BITS67
3070    522D    CNVR 123    X10PLS                         V0=V0L
3072    6223    CNVR 124                                   V0=V0+V0            X2
3074    5239    CNVR 125                                   V1=V0
3076    6223    CNVR 126                                   V0=V0+V0            X4
3078    6223    CNVR 127                                   V0=V0+V0            X8
307A    6233    CNVR 128                                   V0=V0+V1            X10
307C    62C3    CNVR 129                                   V0=V0+P0            PLUS ADDATIVE
307E    128E    CNVR 130                                   RTN
                                                  **********************************
                                                  * CROSS REFERENCE FOR CSECT CNVR *
                                                  **********************************

CNVR 005     CICY 199  CICY 200
CNVR 009     CNVR 007
CNVR 013     CNVR 011
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT CNVR *
                                  **********************************

CNVR 019     CNVR 016
CNVR 024     CNVR 025
CNVR 026     CNVR 023  CNVR 027
CNVR 027     CNVR 029
CNVR 030     CNVR 026
CNVR 031     CNVR 028
CNVR 039     CNVR 024
CNVR 046     CNVR 031  CNVR 043
CNVR 051     CNVR 049
CNVR 053     CNVR 051
CNVR 058     CNVR 012
CNVR 061     CNVR 067
CNVR 065     CNVR 063
CNVR 074     CNVR 072
CNVR 080     CNVR 078
CNVR 081     CNVR 099
CNVR 085     CNVR 107
CNVR 098     CNVR 093
CNVR 100     CNVR 090
CNVR 106     CNVR 103  CNVR 109
CNVR 108     CNVR 104
CNVR 110     CNVR 097
CNVR 116     CNVR 112
CNVR 117     CNVR 113  CNVR 115
CNVR 120     CDVD 099  CNVR 114  CNVR 116
CNVR 122     CNVR 119
CNVR 123     CNVR 087  CNVR 101
```

CSAS -- DECIMAL ROUTINES

ENTRIES
1. SSOPF8 -- ZERO AND ADD
2. SSOPF9 -- COMPARE DECIMAL
3. SSOPFA -- ADD DECIMAL
4. SSOPFB -- SUBTRACT DECIMAL

OBJECTIVES

DECIMAL ADD OR SUBTRACT

THE 2ND OPERAND, CALLED THE SOURCE FIELD, IS EITHER TRUE ADDED OR COMPLEMENT ADDED TO THE 1ST OPERAND, WHICH IS CALLED THE DESTINATION FIELD. THE OPERAND SIGNS AND OP CODE DETERMINE WHETHER TRUE OR COMPLEMENT ADD IS PERFORMED AS FOLLOWS.

SIGN		TRUE OR COMPLEMENT	
1ST OPERAND	2ND OPERAND	ADD OP	SUBTRACT OP
PLUS	PLUS	TRUE	COMPLEMENT
MINUS	PLUS	COMPLEMENT	TRUE
MINUS	MINUS	TRUE	COMPLEMENT
PLUS	MINUS	COMPLEMENT	TRUE

ZERO AND ADD

THE 2ND OPERAND IS ADDED TO A VALUE OF ZERO AND PLACED IN THE 1ST OPERAND LOCATION.

COMPARE DECIMAL

THE 2ND OPERAND IS COMPARED WITH THE 1ST OPERAND, AND THE RESULT OF THE COMPARISON IS INDICATED BY THE CONDITION CODE. COMPARISON IS RIGHT TO LEFT.

DESCRIPTION

THE CSAS ROUTINE PERFORMS THE DECIMAL OPERATIONS AS FOLLOWS.
1. FOR ADD OR ZERO AND ADD, SET TRUE ADD.
   FOR SUBTRACT OR COMPARE, SET COMPLEMENT ADD.
2. READ LOW-ORDER BYTE OF 2ND OPERAND. TEST SIGN. IF SIGN IS INVALID, INITIATE PROGRAM INTERRUPT.
3. LOOP BACK AND TEST SIGN OF 1ST OPERAND. THIS STEP IS NOT PERFORMED FOR ZERO AND ADD.
4. ADD OR COMPLEMENT ADD 1ST DIGIT.
5. SET PROPER SIGN. STORE 1ST BYTE AT 1ST OPERAND, DESTINATION FIELD, THEN STEP TO NEXT DESTINATION BYTE. RESULT IS NOT STORED IF COMPARE OP.

DESCRIPTION CONTINUED

6. READ NEXT SOURCE BYTE FROM 2ND OPERAND.
   READ NEXT DESTINATION BYTE FROM 1ST OPERAND.
   IF ZERO AND ADD, DESTINATION FIELD IS SET TO ZERO.
7. ADD OR COMPLEMENT ADD THE TWO BYTES.
8. STORE RESULT AT 1ST OPERAND DESTINATION FIELD, AND STEP TO NEXT DESTINATION BYTE. RESULT IS NOT STORED IF COMPARE OP.
9. DECREMENT LENGTH.
10. REPEAT STEPS 6 THROUGH 9 UNTIL EITHER DESTINATION OR SOURCE FIELD ENDS.

SOURCE FIELD ENDS BEFORE DESTINATION FIELD

SUPPLY ZEROES AS SOURCE DIGITS AND CONTINUE UNTIL DESTINATION FIELD ENDS.

DESTINATION FIELD ENDS BEFORE SOURCE FIELD

READ OUT REMAINING SOURCE DIGITS. ADD THEM TO ZERO AFTER CHECKING TRUE/COMPLEMENT STATUS AND THE CARRY FROM LAST ADD AS FOLLOWS.
1. COMPLEMENT ADD AND CARRY. IF LAST ADD BEFORE DESTINATION FIELD ENDED WAS ZERO, FURTHER RESULTS MUST BE ZERO, OR AN OVERFLOW IS INDICATED.
   IF PREVIOUS RESULT WAS NOT ZERO, FURTHER RESULTS MUST BE NINES.
2. COMPLEMENT ADD AND NO CARRY. RESULTS MUST BE NINES, OR AN OVERFLOW IS INDICATED.
3. TRUE ADD AND CARRY. INDICATE OVERFLOW.
4. TRUE ADD AND NO CARRY. INDICATE OVERFLOW IF FURTHER RESULTS ARE NOT ZERO.

BOTH SOURCE AND DESTINATION FIELD ENDED

1. IF INVALID DATA, INITIATE PROGRAM INTERRUPT.
2. RECOMPLEMENT RESULT IF THE FOLLOWING CONDITIONS ARE PRESENT WHEN FIELDS END.
   A. COMPLEMENT ADD, NO CARRY, AND NOT A COMPARE OP.
   B. TRUE ADD, NO CARRY, ALU ZERO, MINUS SIGN STATUS, AND NOT A COMPARE OP.
   RESULT IS RECOMPLEMENTED, STARTING AT LOW-ORDER BYTE. SIGN OF RESULT IS INVERTED.
3. SET CONDITION CODES AND RETURN TO I-CYCLES.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              CSAS 001    T                DECIMAL ADD, SUBTRACT, COMPARE, AND ZERO AND ADD
              CSAS 002    *
              CSAS 003    *                OPERAND 1 ADDRESS IS IN U    H1 IS ZERO    T0 IS ZERO
              CSAS 004    *                OPERAND 2 ADDRESS IS IN V    D1 IS ZERO
              CSAS 005    ASEQ    AL07=4C
134C    3443  CSAS 006     SSOPF8                    G0=G0$K04             ZERO AND ADD
134E    1082  CSAS 007     SSOPFA                    RST S K=98                RESET S0,S3,S4   ADD
1350    2E15  CSAS 008     STUPDC                    H0=0$K10              SET UP DECREMENT
1352    573A  CSAS 009                               RDB D1 V-1            READ 1ST 2ND OP BYTE
1354    576D  CSAS 010     TSTSGN                    D0=D1L
1356    0661  CSAS 011                               Z=D0+K06
1358    E0F3  CSAS 012                  025  NOSIGN   BR IF HZ=0
135A    00C4  CSAS 013                               RST S K=2C                RESET S2,S4,S5
135C    F366  CSAS 014                  019  ISPLUS   BR IF D1 BIT7=0       TEST FOR PLUS  A,C,E
135E    07FB  CSAS 015                               Z=D1<K0F              TEST FOR F
1360    F0E7  CSAS 016                  019  ISPLUS   BR IF DYN BIT7=1      TEST FOR PLUS
1362    C5F5  CSAS 017                  026  ISCOMP   BR IF S0=1            TEST FOR COMP ADD
1364    3042  CSAS 018                               SET S K=94                SET S0,S3,S5
1366    C07C  CSAS 019     ISPLUS       030  ENDSGN   BR IF G0 BIT4=0       SIGN TEST IS OVER
1368    D07B  CSAS 020                  029  ZAPFST   BR IF G0 BIT5=1       BR IF ZERO AND ADD
136A    148B  CSAS 021                               G0=G0<K08             INVERT G04
136C    57FB  CSAS 022                               H1=D1H                STRIP SIGN
136E    5710  CSAS 023                               RDB D1 U+0            READ 2ND OPERAND BYTE
1370    9354  CSAS 024                  010  TSTSGN   BR
1372    9BDC  CSAS 025     NOSIGN   CCOM 071  FLGDEX   BR
1374    1002  CSAS 026     ISCOMP                    RST S K=90                SET UP FOR TRUE ADD
1376    2040  CSAS 027                               SET S5
1378    9366  CSAS 028                  019  ISPLUS   BR
137A    1002  CSAS 029     ZAPFST                    RST S K=90                RESET S0,S3
137C    577B  CSAS 030     ENDSGN                    D1=D1H                STRIP SIGN
137E    77FF  CSAS 031                               D1C=D1@H1+C           ADD FIRST CHAR
1380    C905  CSAS 032                  034  ISASCI   BR IF P1 BIT4=1       BR IF ASCII
1382    272B  CSAS 033                               D1=D1+K02
1384    D188  CSAS 034     ISASCI       036  DSPLUS   BR IF S5=0            BR IF PLUS
1386    271B  CSAS 035                               D1=D1+K01
1388    27AB  CSAS 036     DSPLUS                    D1=D1+K0A
138A    F035  CSAS 037                  058  DECCMP   BR IF G0 BIT7=1       TEST FOR COMPARE
138C    771A  CSAS 038                               STB D1 U-1
138E    55B9  CSAS 039     TSTLGT                    T1=G1
1390    E0EB  CSAS 040                  085  DESTND   BR IF HZ=0            TEST FOR DEST END
1392    F0A6  CSAS 041                  051  MORSOR   BR IF LZNZ            TEST FOR SOURCE LEFT
1394    2707  CSAS 042     ENDSOR                    D1=0
1396    2EF3  CSAS 043                               H0=0$K0F
1398    D02A  CSAS 044                  053  RDDEST   BR IF G0 BIT5=0
139A    2F07  CSAS 045     ZRODST                    H1=0
139C    93AC  CSAS 046                  054  COMPUT   BR
139E    7F1A  CSAS 047     STORED                    STB H1 U-1
13A0    75E3  CSAS 048     DECLGT                    G1=G1-H0
13A2    E0EB  CSAS 049                  085  DESTND   BR IF HZ=0
13A4    F095  CSAS 050                  042  ENDSOR   BR IF LZ=0
13A6    573A  CSAS 051     MORSOR                    RDB D1 V-1
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
13A8  D01B  CSAS 052                 045    ZRODST     BR IF GO BIT5=1
13AA  5F1C  CSAS 053     RDDEST                        RDB H1 U+0
13AC  7F7F  CSAS 054     COMPUT                        H1C=H1@D1+C
13AE  F01E  CSAS 055                 047    STORED     BR IF GO BIT7=0
13B0  5006  CSAS 056                                   U=U-1
13B2  93A0  CSAS 057                 048    DECLGT     BR
13B4  5006  CSAS 058     DECCMP                        U=U-1
13B6  938E  CSAS 059                 039    TSTLGT     BR
13B8  1413  CSAS 060     SSOPFB                        GO=GC*-K01         SUBTRACT  RST G07
13BA  3002  CSAS 061     SSOPF9                        SET S K=90             COMPARE   SET COMP&S3
13BC  9350  CSAS 062                 008    STUPDC     BR
13BE  573A  CSAS 063     COMPAD                        RDB D1 V-1
13C0  F5D1  CSAS 064                 072    COMPGD     BR IF S3=1
13C2  7A7F  CSAS 065                                   TOC=TC@D1+C
13C4  1A9F  CSAS 066     TSTNIN                        TO=TO<K99
13C6  C4F9  CSAS 067                 092    OKSOFA     BR IF Z=0
13C8  0002  CSAS 068                                   RST S3
13CA  1415  CSAS 069     SETOVF                        GO=GC*-K10
13CC  2A07  CSAS 070                                   TO=0
13CE  93F8  CSAS 071                 092    OKSOFA     BR
13D0  E5D9  CSAS 072     COMPGD      076    MYBEOK     BR IF S2=1
13D2  7A7F  CSAS 073                                   TOC=TO@D1+C
13D4  C4F9  CSAS 074                 092    OKSOFA     BR IF Z=0
13D6  93CA  CSAS 075                 069    SETOVF     BR
13D8  7A7F  CSAS 076     MYBEOK                        TOC=TO@D1+C
13DA  F5E1  CSAS 077                 080    STLOKA     BR IF S3=1
13DC  E445  CSAS 078                 066    TSTNIN     BR IF GO BIT2=1
13DE  93CA  CSAS 079                 069    SETOVF     BR
13E0  142D  CSAS 080     STLOKA                        GO=GO<K20
13E2  93F8  CSAS 081                 092    OKSOFA     BR
13E4  77AF  CSAS 082     TRUCAR                        D1C=D1@TO+C
13E6  93F6  CSAS 083                 091    OVFLAA     BR
13E8  9372  CSAS 084     NOOSIN      025    NOSIGN     BR
13EA  F0FD  CSAS 085     DESTND      094    ALLEND     BR IF LZ=0
13EC  C5BF  CSAS 086     MORDST      063    COMPAD     BR IF SO=1
13EE  573A  CSAS 087                                   RDB D1 V-1
13F0  F5E5  CSAS 088                 082    TRUCAR     BR IF S3=1
13F2  77AF  CSAS 089                                   D1C=D1@TO+C
13F4  C4F9  CSAS 090                 092    OKSOFA     BR IF Z=0
13F6  2002  CSAS 091     OVFLAA                        SET S3
13F8  25FF  CSAS 092     OKSOFA                        G1=G1+KFF
13FA  F0EC  CSAS 093                 086    MORDST     BR IF LZNZ
13FC  C1E9  CSAS 094     ALLEND      084    NOOSIN     BR IF S4=1
13FE  C591  CSAS 095     ALENDH      104    ALENDB     BR IF SO=1
1400  F589  CSAS 096                 100    ALENDA     BR IF S3=1
1402  E58B  CSAS 097     ALENDE      101    ALENDC     BR IF S2=1
1404  D1D1  CSAS 098                 136    ALENDI     BR IF S5=1
1406  AA74  CSAS 099     ALENDJ  CICY 003   CHECKX     BR                                  0
1408  F03C  CSAS 100     ALENDA      126    ALENDG     BR IF GO BIT7=0
140A  D18F  CSAS 101     ALENDC      103    ALENDD     BR IF S5=1
140C  A824  CSAS 102             CLOG 063   OP95A      BR                                  2
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT              COMMENTS
140E    A820    CSAS 103    ALENDD    CLOG 061   OP95C  0   BR                      1
1410    F583    CSAS 104    ALENDB         097   ALENDE     BR IF S3=1
1412    F047    CSAS 105                   131   ALENDF     BR IF G0 BIT7=1
1414    5BB5    CSAS 106    RECOMP                          T1=T1XL
1416    61B9    CSAS 107                                    U1C=U1+T1+1
1418    60AD    CSAS 108                                    U0C=U0+T0+C
141A    5710    CSAS 109                                    RDB D1 U+0
141C    57FB    CSAS 110                                    H1=D1H
141E    3002    CSAS 111                                    SET S K=90
1420    577D    CSAS 112                                    D1=D1L
1422    171B    CSAS 113                                    D1=D1<K01
1424    7AFF    CSAS 114                                    T0C=T0@H1+C
1426    67A3    CSAS 115                                    D1=D1+T0
1428    771A    CSAS 116                                    STB D1 U-1
142A    5BA9    CSAS 117                                    T0=T1
142C    C4BB    CSAS 118                   125   RCMPOV     BR IF Z=0
142E    5710    CSAS 119    NXTCHR                          RDB D1 U+C
1430    757F    CSAS 120                                    G1C=G1@D1+C
1432    751A    CSAS 121                                    STB G1 U-1
1434    2507    CSAS 122                                    G1=0
1436    2AFF    CSAS 123                                    T0=T0+KFF
1438    C4AF    CSAS 124                   119   NXTCHR     BR IF ZNZ
143A    F447    CSAS 125    RCMPOV         131   ALENDF     BR IF GC3=1
143C    2C75    CSAS 126    ALENDG                          P0=0$K70                SET CC 3
143E    DD43    CSAS 127                   129   DEOVFL     BR IF P1 BIT1=1
1440    AA76    CSAS 128               CICY 004   CHECK      BR
1442    2BA3    CSAS 129    DEOVFL                          T1=0$K0A                DECIMAL OVERFLOW
1444    89FC    CSAS 130               BPSW 003   PROG       BR
1446    D1CD    CSAS 131    ALENDF         134   INVSNB     BR IF S5=1
1448    2040    CSAS 132                                    SET S5
144A    9402    CSAS 133                   097   ALENDE     BR
144C    0040    CSAS 134    INVSNB                          RST S5
144E    9402    CSAS 135                   097   ALENDE     BR
1450    F007    CSAS 136    ALENDI         099   ALENDJ     BR IF G0 BIT7=1
1452    9414    CSAS 137                   106   RECOMP     BR
                CSAS 138    AEND
```

****
* CROSS REFERENCE FOR CSECT CSAS *
****

```
CSAS 006    CICY 280
CSAS 007    CICY 282
CSAS 008    CSAS 062
CSAS 010    CSAS 024
CSAS 019    CSAS 014  CSAS 016  CSAS 028
CSAS 025    CSAS 012  CSAS 084
CSAS 026    CSAS 017
CSAS 029    CSAS 020
CSAS 030    CSAS 019
CSAS 034    CSAS 032
CSAS 036    CSAS 034
CSAS 039    CSAS 059
```

****
* CROSS REFERENCE FOR CSECT CSAS *
****

CSAS 042 CSAS 050
CSAS 045 CSAS 052
CSAS 047 CSAS 055
CSAS 048 CSAS 057
CSAS 051 CSAS 041
CSAS 053 CSAS 044
CSAS 054 CSAS 046
CSAS 058 CSAS 037
CSAS 060 CICY 283
CSAS 061 CICY 281
CSAS 063 CSAS 086
CSAS 066 CSAS 078
CSAS 069 CSAS 075 CSAS 079
CSAS 072 CSAS 064
CSAS 076 CSAS 072
CSAS 080 CSAS 077
CSAS 082 CSAS 088
CSAS 084 CSAS 094
CSAS 085 CSAS 040 CSAS 049
CSAS 086 CSAS 093
CSAS 091 CSAS 083
CSAS 092 CSAS 067 CSAS 071 CSAS 074 CSAS 081 CSAS 090
CSAS 094 CSAS 085
CSAS 097 CSAS 104 CSAS 133 CSAS 135
CSAS 099 CSAS 136
CSAS 100 CSAS 096
CSAS 101 CSAS 097
CSAS 103 CSAS 101
CSAS 104 CSAS 095
CSAS 106 CSAS 137
CSAS 119 CSAS 124
CSAS 125 CSAS 118
CSAS 126 CSAS 100
CSAS 129 CSAS 127
CSAS 131 CSAS 105 CSAS 125
CSAS 134 CSAS 131
CSAS 136 CSAS 098

CSFT -- SHIFT ROUTINES

ENTRIES
1. RSOP89 -- SHIFT LEFT LOGICAL
   RSOP8B -- SHIFT LEFT ALGEBRAIC
   RSOP8D -- SHIFT LEFT DOUBLE LOGICAL
   RSOP8F -- SHIFT LEFT DOUBLE ALGEBRAIC
   RSOP88 -- SHIFT RIGHT LOGICAL
   RSOP8A -- SHIFT RIGHT ALGEBRAIC
   RSOP8C -- SHIFT RIGHT DOUBLE LOGICAL
   RSOP8E -- SHIFT RIGHT DOUBLE ALGEBRAIC

OBJECTIVES -- ALL SHIFTS

OBTAIN BINARY VALUE OF THE 2ND OPERAND EFFECTIVE ADDRESS LOW-ORDER 6 BITS. THEN SHIFT THE 1ST OPERAND EITHER LEFT OR RIGHT THAT AMOUNT.

DESCRIPTION

THE OPERANDS ARE FETCHED, THEN THE OP CODE IS TESTED. FOR ALGEBRAIC SHIFTS WITH NEGATIVE OPERAND, A REGISTER IS SET TO FF FOR EXTENDING THE SIGN. THE HIGH-ORDER BITS OF THE SHIFT-COUNT FIELD ARE HANDLED FIRST. THESE BITS REPRESENT SHIFTS OF 32, 16, AND 8. THE -SHIFT8- SUBROUTINE IS EXECUTED THE NUMBER OF TIMES NEEDED TO ACCOMPLISH ANY OF THE HIGH-ORDER SHIFTS.

AFTER THE HIGH-ORDER SHIFTS ARE COMPLETED, THE OP CODE IS TESTED FOR RIGHT OR LEFT SHIFT. THEN THE SHIFTS FOR THE LOW-ORDER 3 BITS OF THE COUNT FIELD ARE HANDLED. THESE BITS REPRESENT SHIFTS OF 4, 2, AND 1.

LEFT SHIFT 1 THROUGH 7. THE -LEFT1- SUBROUTINE IS USED. THIS SUBROUTINE SHIFTS 1 BIT, THEN DECREMENTS THE LOW-ORDER COUNT. THE SUBROUTINE LOOPS UNTIL THE COUNT IS REDUCED TO ZERO.

RIGHT SHIFT 1 THROUGH 7. IF A RIGHT SHIFT OF 4 IS NEEDED, IT IS HANDLED BY THE --RIGHT4- SUBROUTINE. THEN THE BITS REPRESENTING 8 AND 4 ARE RESET IN THE COUNT. THE BITS REPRESENTING 2 AND 1, FOR SHIFTS OF 0 THROUGH 3 ARE HANDLED AS FOLLOWS.

LOW 2 BITS OF COUNT		
00	DONE.	GO TO SHIFT COMPLETE
01	SET COUNT TO 3.	RIGHT 1 EQUALS 3 LEFT MINUS 4 RIGHT.
10	COUNT OKAY.	RIGHT 2 EQUALS 2 LEFT MINUS 4 RIGHT.
11	SET COUNT TO 1.	RIGHT 3 EQUALS 1 LEFT MINUS 4 RIGHT.

DESCRIPTION CONTINUED

AFTER THE COUNT IS SET UP, THE -LEFT1- SUBROUTINE LOOPS UNTIL THE COUNT IS REDUCED TO ZERO. THEN THE RIGHT SHIFT OF 4 IS DONE BY THE -RIGHT4- SUBROUTINE.

SHIFT COMPLETE. THIS SECTION OF THE ROUTINE STORES THE RESULT BACK INTO THE 1ST OPERAND. FOR LOGICAL SHIFTS, THE MICROPROGRAM RESTORES SAVED REGISTERS AND RETURNS TO I-CYCLES. FOR ALGEBRAIC SHIFTS, SO IS TESTED TO DETERMINE IF AN OVERFLOW OCCURRED. FOR SHIFT OVERFLOW, THE SIGN IS INVERTED AND STORED, AND THE MICROPROGRAM EXITS TO SET CONDITION CODE 3. IF NO OVERFLOW, THE MICROPROGRAM SETS THE CONDITION CODE ACCORDING TO THE SIGN AND WHETHER THE RESULT IS ZERO OR NON-ZERO. THE SAVED REGISTERS ARE RESTORED, AND THE MICROPROGRAM RETURNS TO I-CYCLES.

SUBROUTINES

-SHIFT8-. THIS SUBROUTINE HANDLES BOTH RIGHT AND LEFT SHIFTS. SHIFT RIGHT 8 IS ACCOMPLISHED BY A STRAIGHT MOVE OF EACH BYTE-SOURCE REGISTER TO THE NEXT ADJACENT REGISTER ON THE RIGHT. VACATED HIGH-ORDER BITS ARE FILLED FROM A REGISTER THAT CONTAINS BITS LIKE THE ORIGINAL SIGN.

SHIFT LEFT 8 IS ACCOMPLISHED BY A STRAIGHT MOVE OF EACH BYTE-SOURCE REGISTER TO THE NEXT ADJACENT REGISTER ON THE LEFT. ANY CHANGE IN SIGN IS AN OVERFLOW AND MUST BE REMEMBERED TO SET CONDITION CODE 3 FOR ALGEBRAIC SHIFTS.

-LEFT1-. SHIFT LEFT 1 IS ACCOMPLISHED BY ADDING EACH BYTE TO ITSELF, SAVING CARRIES. SIGN CHANGE IS STORED BY SETTING SO TO INDICATE OVERFLOW.

-RIGHT4-. EACH HALF BYTE IS CROSSED INTO THE NEXT ADJACENT HALF-BYTE ON THE RIGHT.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		CSFT 001	T			SHIFT ROUTINES	
		CSFT 002	*				
		CSFT 003	*				

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                CSFT 004    *
                CSFT 005    *                              ALL 8 SHIFTS START AT THE FIRST WORD OF THIS ROUTINE
                CSFT 006    *                              DOUBLE SHIFT DATA  IS IN    H,D,V,U
                CSFT 007    *                              SINGLE SHIFT DATA  IS IN    H,D
                CSFT 008    *                              ORIGINAL SIGN EXPANDED TO 8 BITS =TC
                CSFT 009    *                              COUNT FOR LOOPS  IS IN           PO OR P1
                CSFT 010    *                              NO OF BIT POSITIONS TO BE SHIFTED IS IN P1
                CSFT 011    *                              S5=ORIG SIGN-  , SO USED FOR OVERFLOW
                CSFT 012    *                                  **********
                CSFT 013    *
1CFA  10CE      CSFT 014     SHIFT                         RST S K=FC
1CFC  7CC2      CSFT 015                                   STH P DA,B8
1CFE  51D9      CSFT 016                                   P1=U1                  SAVE SHIFT COUNT
1D00  55BB      CSFT 017                                   T1=G1H                 GET R1
1D02  5EA8      CSFT 018                                   RDH H AS,T+2
1D04  56AA      CSFT 019                                   RDH D AS,T-2
1D06  D014      CSFT 020                    027   START    BR IF G05=0            BR IF SHORT
1D08  F50E      CSFT 021                    024   NOSP     BR IF G13=0            BR IF R1 EVEN
1D0A  5CC2      CSFT 022                                   RDH P DA,B8            RESTORE P
1D0C  AE94      CSFT 023              CICY 070  SPECHK     BR
1D0E  3B15      CSFT 024     NOSP                          T1=T1$K10              GET SECOND HALF
1D10  52A8      CSFT 025                                   RDH V AS,T+2            OF
1D12  50AA      CSFT 026                                   RDH U AS,T-2             DOUBLE  OPERAND
1D14  E01C      CSFT 027     START          031   LOGPOS   BR IF G06=0            BR IF LOGICAL OP
1D16  CE1C      CSFT 028                    031   LOGPOS   BR IF H00=0            BR IF +
1D18  2AF7      CSFT 029                                   T0=0$KFF               EXTEND - SIGN
1D1A  2040      CSFT 030                                   SET S5                 REMEMBER NEG
1D1C  5DC9      CSFT 031      LOGPOS                       P0=P1                  MOVE SHIFT COUNT
1D1E  6CC3      CSFT 032                                   P0=P0+P0               MOVE THE 32,16,8 CTS LEFT ONE
1D20  1C85      CSFT 033                                   P0=P0*-K80
1D22  E0AB      CSFT 034                    038   NO8      BR IF HZ=0             BR IF NO 32,16 OR 8 SHIFT
1D24  B29A      CSFT 035     BAL            129   SHIFT8   BAL                    SHIFT ONE  BYTE LEFT OR RIGHT
1D26  2CFD      CSFT 036                                   P0=P0+KF0              DEC HI COUNT
1D28  E0A4      CSFT 037                    035   BAL      BR IF HZNZ
1D2A  F035      CSFT 038     NO8            054   LEFT     BR IF G07=1            BREAK OUT LEFT SHIFTS
1D2C  D930      CSFT 039                    041   NO4      BR IF P15=0            BR IF NO RIGHT 4
1D2E  B340      CSFT 040                    107   RIGHT4   BAL
1D30  1DC3      CSFT 041     NO4                          P1=P1*-K0C
1D32  8977      CSFT 042                    046   LOW    N N=P1 BITS67
                CSFT 043    *                               **********
                CSFT 044    *              RIGHT 1=3L-4R  **  RIGHT 2=2L-4R **  RIGHT 3=1L-4R
                CSFT 045    *                               LOW ORDER BITS MUST BE CORRECTED
1BE0  9D3A      CSFT 046     LOW    0       057   SHFTCC   BR                     NO 1,2 OR 3 RIGHT SHIFT
1BE2  2D28      CSFT 047     LOW    1                      P1=P1+K02              R1=3L-4R   FIX P1=3
1BE4  9BE8      CSFT 048     LOW    2       050   BALRL    BR                     R2=2L-4R  P1=OK
1BE6  1D23      CSFT 049     LOW    3                      P1=P1*-K02             R3=1L-4R   FIX P1
1BE8  8420      CSFT 050     BALRL          087   LEFT1    BAL
1BEA  B340      CSFT 051                    107   RIGHT4   BAL
1BEC  9D3A      CSFT 052                    057   SHFTCC   BR
                CSFT 053    *                               **********
1D34  1D83      CSFT 054     LEFT                         P1=P1*-K08             REMOVE HI BIT
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
1D36   F0BB    CSFT 055                 057   SHFTCC     BR IF LZ=0
1D38   8420    CSFT 056                 087   LEFT1      BAL                DO LEFT SHIFT 1 THRU 7 PLACES
1D3A   2A07    CSFT 057    SHFTCC                        T0=0               REMOVE EXT SIGN
1D3C   5CC2    CSFT 058                                  RDH P DA,B8        RESTORE MASK
1D3E   D046    CSFT 059                 063   SHST       BR IF GC5=0
1D40   72A8    CSFT 060                                  STH V AS,T+2       STORE
1D42   70AA    CSFT 061                                  STH U AS,T-2        LOW
1D44   1B15    CSFT 062                                  T1=T1*-K10           DOUBLE
1D46   7EA8    CSFT 063    SHST                          STH H AS,T+2       STORE
1D48   76AA    CSFT 064                                  STH D AS,T-2        SINGLE OR HI DOUBLE
1D4A   E04F    CSFT 065                 067   ALG        BR IF GC6=1
1D4C   AA76    CSFT 066               CICY 004  CHECK     BR                 LOGICAL GO BACK
1D4E   F05E    CSFT 067    ALG          075   NOINT      BR IF GC7=0        BR IF RIGHT-NO OVF CHECK
1D50   C5DE    CSFT 068                 075   NOINT      BR IF S0=0         LEFT ALG CHECK OVF
1D52   56A0    CSFT 069                                  RDH D AS,T+0       HAVE OVF FIX SIGN
1D54   1685    CSFT 070                                  D0=D0*-K80         MAKE SIGN +
1D56   D1DA    CSFT 071                 073   CSIGN      BR IF S5=0         BR IF ORIGINAL SIGN +
1D58   3685    CSFT 072                                  D0=D0$K80          MAKE SIGN -
1D5A   76A0    CSFT 073    CSIGN                         STH D AS,T+0        STOR CORRECT SIGN
1D5C   AC40    CSFT 074               CLST 011  SHFTOV    BR
1D5E   CE64    CSFT 075    NOINT        077   POS        BR IF H00=0
1D60   A820    CSFT 076               CLOG 061  OP95C  0  BR                 CC 1
1D64   C008    CSFT 077    POS                           RST S1
1D66   4EE6    CSFT 078                                  H=H                 TEST
1D68   4666    CSFT 079                                  D=D                  HI
1D6A   D070    CSFT 080                 083   SHORT      BR IF GC5=0        BR IF SINGLE
1D6C   4226    CSFT 081                                  V=V                   TEST
1D6E   4006    CSFT 082                                  U=U                    LOW DOUBLE
1D70   D5E3    CSFT 083     SHORT       085   CC2        BR IF S1=1
1D72   AA74    CSFT 084               CICY 003  CHECKX    BR                 CCO
1D62   A824    CSFT 085     CC2       CLOG 063  OP95A     BR
               CSFT 086    *                              *** LEFT 1 ROUTINE   ***
0420   C002    CSFT 087     LEFT1                         RST S3
0422   D02C    CSFT 088                 093   SHORT1     BR IF GC5=0
0424   611B    CSFT 089                                  U1C=U1+U1             SHIFT
0426   600D    CSFT 090                                  U0C=U0+U0+C
0428   633D    CSFT 091                                  V1C=V1+V1+C             LEFT
042A   622D    CSFT 092                                  V0C=V0+V0+C
042C   677D    CSFT 093     SHORT1                       D1C=D1+D1+C               ONE
042E   666D    CSFT 094                                  D0C=D0+D0+C
0430   6FFD    CSFT 095                                  H1C=H1+H1+C                BIT
0432   6EED    CSFT 096                                  H0C=H0+H0+C
0434   E4B8    CSFT 097                 099   SKIPST     BR IF NOVFL
0436   3000    CSFT 098                                  SET S0
0438   6AAD    CSFT 099     SKIPST                       T0C=T0+T0+C           SAVE CARRY FOR RIGHT SHIFTS
043A   2DFB    CSFT 100                                  P1=P1+K0F             DEC COUNT
043C   F0A0    CSFT 101                 087   LEFT1      BR IF LZNZ
043E   128E    CSFT 102                                  RTN
               CSFT 103    *
               CSFT 104    *
               CSFT 105    *
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS

              CSFT 106    *                                  *** RIGHT  4  ROUTINE ***
3340    D052  CSFT 107     RIGHT4         116   R4SHOR     BR IF G05=0
3342    5115  CSFT 108                                      U1=U1XL
3344    4013  CSFT 109                                      U1=U0XH+U1L
3346    5005  CSFT 110                                      U0=U0XL
3348    4303  CSFT 111                                      U0=V1XH+U0L
334A    5335  CSFT 112                                      V1=V1XL
334C    4233  CSFT 113                                      V1=V0XH+V1L
334E    5225  CSFT 114                                      V0=V0XL
3350    4723  CSFT 115                                      V0=D1XH+V0L
3352    5775  CSFT 116     R4SHOR                           D1=D1XL
3354    4673  CSFT 117                                      D1=D0XH+D1L
3356    5665  CSFT 118                                      D0=D0XL
3358    4F63  CSFT 119                                      D0=H1XH+D0L
335A    5FF5  CSFT 120                                      H1=H1XL
335C    4EF3  CSFT 121                                      H1=H0XH+H1L
335E    5EE5  CSFT 122                                      H0=H0XL
3360    4AE3  CSFT 123                                      H0=T0XH+H0L
3362    128E  CSFT 124                                      RTN
              CSFT 125    *
              CSFT 126    *
              CSFT 127    *
              CSFT 128    *                                  ** COMMON SHIFT 8    RIGHT AND LEFT  ***
329A    F001  CSFT 129     SHIFT8         140   LEFT8      BR IF G07=1           BR IF LEFT SHIFT
329C    D026  CSFT 130                    135   SHORTR     BR IF G05=0           THIS IS RIGHT8
329E    5019  CSFT 131                                      U1=U0                  SHIFT
32A0    5309  CSFT 132                                      U0=V1
32A2    5239  CSFT 133                                      V1=V0                    RIGHT
32A4    5729  CSFT 134                                      V0=D1
32A6    5679  CSFT 135     SHORTR                           D1=D0                      ONE
32A8    5F69  CSFT 136                                      D0=H1
32AA    5EF9  CSFT 137                                      H1=H0                        BYTE
32AC    5AE9  CSFT 138                                      H0=T0                 SHIFT
32AE    128E  CSFT 139                                      RTN
3280    6EA1  CSFT 140     LEFT8                            H0=H0<T0              CHECK HI 8 BITS FOR SIGN CHANGE
3282    C48C  CSFT 141                    146   OVFSET     BR IF ZNZ
3284    D18A  CSFT 142                    145   PLUS       BR IF S5=0
3286    CF0F  CSFT 143                    147   OK         BR IF H10=1           ORIG SIGN-BR IF NEW SIGN-
3288    3000  CSFT 144                                      SET S0                ORIG SIGN-NEW SIGN+REMBER OVF
328A    CF0E  CSFT 145     PLUS           147   OK         BR IF H10=0           BR IF NEW SIGN BIT +
328C    300C  CSFT 146     OVFSET                           SET S0                REMEMBER OVERFLOW
328E    5FE9  CSFT 147     OK                               H0=H1                  SHIFT
3290    56F9  CSFT 148                                      H1=D0
3292    5769  CSFT 149                                      D0=D1                    LEFT
3294    D031  CSFT 150                    153   LONGL8     BR IF G05=1
3296    2707  CSFT 151                                      D1=0                      ONE
3298    128E  CSFT 152                                      RTN
32B0    5279  CSFT 153     LONGL8                           D1=V0                        BYTE
32B2    5329  CSFT 154                                      V0=V1
32B4    5039  CSFT 155                                      V1=U0
32B6    5109  CSFT 156                                      U0=U1
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

3288    2107    CSFT 157                                     U1=0
328A    128E    CSFT 158                                     RTN
                                                  ************************************
                                                  * CROSS REFERENCE FOR CSECT CSFT *
                                                  ************************************

CSFT 014    CICY 061
CSFT 024    CSFT 021
CSFT 027    CSFT 020
CSFT 031    CSFT 027  CSFT 028
CSFT 035    CSFT 037
CSFT 038    CSFT 034
CSFT 041    CSFT 039
CSFT 046    CSFT 042
CSFT 050    CSFT 048
CSFT 054    CSFT 038
CSFT 057    CSFT 046  CSFT 052  CSFT 055
CSFT 063    CSFT 059
CSFT 067    CSFT 065
CSFT 073    CSFT 071
CSFT 075    CSFT 067  CSFT 068
CSFT 077    CSFT 075
CSFT 083    CSFT 080
CSFT 085    CSFT 083
CSFT 087    CSFT 050  CSFT 056  CSFT 101
CSFT 093    CSFT 088
CSFT 099    CSFT 097
CSFT 107    CSFT 040  CSFT 051
CSFT 116    CSFT 107
CSFT 129    CSFT 035
CSFT 135    CSFT 130
CSFT 140    CSFT 129
CSFT 145    CSFT 142
CSFT 146    CSFT 141
CSFT 147    CSFT 143  CSFT 145
CSFT 153    CSFT 150
```

CSTS -- STATUS SWITCHING INSTRUCTIONS

ENTRIES

RSOP82 -- LOAD PSW	RSOP93 -- TEST AND SET
RSOP80 -- SET SYSTEM MASK	RROP04 -- SET PROGRAM MASK
RSOP83 -- DIAGNOSE	RROP0A -- SUPERVISOR CALL
RROP08 -- SET STORAGE KEY	RSOP84 -- WRITE DIRECT
RROP08 -- INSERT STORAGE KEY	RSOP85 -- READ DIRECT

OBJECTIVES

GENERAL. THE FOLLOWING TESTS ARE MADE IF APPLICABLE TO INSTRUCTION BEING EXECUTED.

1. CHECK THAT ADDRESS DOES NOT EXCEED 16 BITS.
2. IF PRIVILEGED OP, TEST FOR SUPERVISOR STATE.
3. CHECK ADDRESS BOUNDARY.

LOAD PSW

THE DOUBLE-WORD AT THE LOCATION DESIGNATED BY THE EFFECTIVE OPERAND ADDRESS REPLACES THE PSW.
THE MICROROUTINE TESTS FOR ERRORS, THEN BRANCHES TO THE -BPSW- ROUTINE TO LOAD THE PSW.

SET SYSTEM MASK

THE BYTE AT THE 2ND OPERAND EFFECTIVE ADDRESS REPLACES THE SYSTEM MASK, BITS 0-7, OF THE CURRENT PSW.
AFTER TESTING FOR ERRORS, THE ROUTINE READS THE 2ND OPERAND. THE NEW MASK BYTE IS PLACED IN THE SYSTEM MASK REGISTER -SM- AND IN AUXILIARY STORAGE LOCATION 00A8.

DIAGNOSE

THE DIAGNOSE INSTRUCTION PROVIDES A MEANS OF GOING DIRECTLY TO A SPECIFIC MICROINSTRUCTION. THE ADDRESS OF THIS MICROINSTRUCTION IS GIVEN IN THE OPERAND OF THE DIAGNOSE INSTRUCTION.
THE OPERAND EFFECTIVE ADDRESS IS STORED IN AUXILIARY STORAGE LOCATION 00BC. THEN THIS ADDRESS IS READ FROM AUXILIARY STORAGE INTO THE BACKUP-ZONE I-REG. AFTER THE L.S. ZONE IS SET BACK TO 0, A RETURN WORD IS EXECUTED. THIS CAUSES THE BACKUP I-REG TO BE USED TO ADDRESS THE NEXT MICROINSTRUCTION.

SET STORAGE KEY

SET THE STORAGE-PROTECT KEY FOR THE PROGRAM-STORAGE BLOCK ADDRESSED BY R2.

THE ROUTINE READS THE BLOCK ADDRESS AND KEY. NEXT THE KEY IS PUT IN THE Q-REG USING STP0 MNEMONIC. THEN, THE KEY IS STORED IN STP1 AT THE SPECIFIED BLOCK ADDRESS. BEFORE RETURNING TO I-CYCLES, THE CPU KEY IS RESTORED.

INSERT STORAGE KEY

READ THE STORAGE-PROTECT KEY FOR THE PROGRAM-STORAGE BLOCK ADDRESSED BY R2, AND SET IT IN R1.
THE ROUTINE READS THE BLOCK ADDRESS FROM R2. NEXT, THE KEY IS READ FROM STP1 INTO THE Q-REG. THE KEY IS PUT INTO R1, THEN THE CPU KEY IS RESTORED.

TEST AND SET

TEST THE HIGH-ORDER BIT OF THE BYTE AT THE EFFECTIVE ADDRESS, AND SET THE CONDITION CODE.
THE ROUTINE READS THE ADDRESSED BYTE, THEN TESTS THE HIGH-ORDER BIT FOR 1. IF THE BIT IS 1, CONDITION CODE 1 IS SET. IF THE BIT IS 0, CONDITION CODE 0 IS SET.

SET PROGRAM MASK

BITS 2-7 OF R1 REPLACE THE CONDITION CODE AND THE PROGRAM MASK BITS OF THE CURRENT PSW.
THE CONDITION CODE IS SET IN P0 AND THE PROGRAM MASK IS SET IN P1.

SUPERVISOR CALL

THIS INSTRUCTION CAUSES A SUPERVISOR-CALL INTERRUPTION. THE INTERRUPT CODE IS SUPPLIED IN BYTE 2 OF THE INSTRUCTION.
THE MICROPROGRAM SETS UP A POINTER TO THE SVC OLD PSW, THEN BRANCHES TO THE LOAD PSW ROUTINE.

WRITE DIRECT

THE BYTE AT THE LOCATION DESIGNATED BY THE OPERAND ADDRESS IS MADE AVAILABLE AS A SET OF DIRECT-OUT STATIC SIGNALS. INSTRUCTION BITS 8-15 ARE MADE AVAILABLE AS SIGNAL-OUT TIMING SIGNALS.

READ DIRECT

A DIRECT-IN DATA BYTE IS ACCEPTED FROM AN EXTERNAL DEVICE IN THE ABSENCE OF A HOLD SIGNAL AND IS PLACED IN THE LOCATION DESIGNATED BY THE OPERAND ADDRESS. INSTRUCTION-BITS 8-15 ARE MADE AVAILABLE AS SIGNAL-OUT TIMING SIGNALS.

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
                CSTS 001  T              STATUS SWITCHING INSTRUCTIONS
                CSTS 002  *
2A24  AB25      CSTS 003   RSOP82   CICY 056  ADDERR N   BR IF H1=NZ        LOAD PSW
2A26  F933      CSTS 004              021  OP80A       BR IF P17=1         OPERATION
2A28  0183      CSTS 005                                 Z=U1*-K08            CODE
2A2A  F0AE      CSTS 006              009  OP82A       BR IF LZNZ             EXECUTION
2A2C  8A26      CSTS 007           BPSW 024  LPSW        BR
                CSTS 008  *
2A2E  AE94      CSTS 009   OP82A    CICY 070  SPECHK      BR
                CSTS 010  *           SET SYSTEM MASK
2A68  AB25      CSTS 011   RSOP80   CICY 056  ADDERR N   BR IF H1=NZ        SET SYSTEM
2A6A  F933      CSTS 012              021  OP80A       BR IF P17=1          MASK
2A6C  5710      CSTS 013                                 RDB D1 U             OP
2A6E  447F      CSTS 014                                 SM=D1                  CODE
2A70  7782      CSTS 015                                 STB D1 DA,A8             EXECUTION
2A72  8185      CSTS 016   OP8IA    CICY 005  ISTART N   N=S BITS67
                CSTS 017  *
                CSTS 018  *           DIAGNOSE INSTRUCTION
2A30  F95E      CSTS 019   RSOP83        034  OP83A       BR IF P17=0
2A32  ADFC      CSTS 021   OP80A    CICY 079  PRIVOP      BR                 PRIV OP
2A5E  70E2      CSTS 034   OP83A                          STH U DA,BC        DIAGNOSE
2A60  2440      CSTS 036                                 SET MODE K=04
2A62  58E2      CSTS 037                                 RDH I DA,BC
2A64  2400      CSTS 038                                 SET MODE K=00
2A66  128E      CSTS 039                                 RTN
                CSTS 040  *           RROPS 08-SET KEY AND 09 INSERT KEY SHARE THIS ROUTINE
2A34  F933      CSTS 044   RROP08        021  OP80A       BR IF P17=1          CHECK PRIV. OP
2A36  55B3      CSTS 045                                 T1=G1XH              PUT R2 ADDRESS IN T1
2A38  5EA8      CSTS 046                                 RDH H AS,T+2
2A3A  AB25      CSTS 047           CICY 056  ADDERR N   BR IF H1=NZ
2A3C  50AA      CSTS 048                                 RDH U AS,T-2         GET BLOCK ADDRESS
2A3E  110D      CSTS 049                                 U1=U1<K00
2A40  F0AE      CSTS 050              009  OP82A       BR IF LZNZ           CHECK FOR SPEC ERROR
2A42  55BB      CSTS 051                                 T1=G1H               GET R1
2A44  3B33      CSTS 052                                 T1=T1$K03            T1=XXXX0011
2A46  F055      CSTS 053              060  INSERT      BR IF G07=1          BR IF INSERT KEY
2A48  57A0      CSTS 054                                 RDB D1 AS,T+0        THIS IS SET KEY-KEY IS IN D1H
2A4A  17F3      CSTS 055                                 D1=D1*-K0F           D1=KKK0000
2A4C  427F      CSTS 056                                 STPO=D1
2A4E  531C      CSTS 057                                 SSK STP1 U+1
2A50  9248      CSTS 058   DONE     CCOM 182  RSTRKY     BAL                  RESTORE CPU KEY
2A52  8185      CSTS 059            CICY 005  ISTART N   N=S BITS67
2A54  521C      CSTS 060   INSERT                        ISK STPO U+2         READ KEY
2A56  527F      CSTS 061                                 D1=STPO
2A58  17F3      CSTS 062                                 D1=D1*-K0F
2A5A  77A0      CSTS 063                                 STB D1 AS,T+0        STORE KEY IN BITS 24-27 OF R1
2A5C  AA50      CSTS 064              058  DONE        BR                   GO TO RESTORE CPU KEY
                CSTS 066  *
                CSTS 067  ATABLE  ADDR=0AD4
0AD4  8ADA      CSTS 068              071  RDSWCH      BAL
0AD6  7098      CSTS 069                                 STH U I+2
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0AD8    AA76    CSTS 070              CICY 004  CHECK      BR
0ADA    500F    CSTS 071    RDSWCH                         U0=SWAB          READ
0ADC    511F    CSTS 072                                   U1=SWCD            SWITCHES
0ADE    128E    CSTS 073                                   RTN                  ROUTINE
                CSTS 074    AEND
                CSTS 075    *             TEST AND SET
2E44    AB25    CSTS 076    RSOP93    CICY 056  ADDERR N  BR IF H1=NZ       TEST
2E46    2FF7    CSTS 077                                   H1=0$KFF           AND
2E48    5710    CSTS 078                                   RDB D1 U             SET
2E4A    7F10    CSTS 079                                   STB H1 U               EXECUTION
2E4C    C752    CSTS 080                  083  OP93A      BR IF D10=0
2E4E    2C55    CSTS 081                                   P0=0$K50
2E50    8185    CSTS 082              CICY 005  ISTART N   N=S BITS67
2E52    2C05    CSTS 083    OP93A                          P0=0$K00
2E54    8185    CSTS 084              CICY 005  ISTART N   N=S BITS67
                CSTS 085    *
                CSTS 086    *             SET PROGRAM MASK
2E32    55BB    CSTS 087    RROP04                         T1=G1H           SET
2E34    5EA0    CSTS 088                                   RDH H AS,T         PROG
2E36    4ED3    CSTS 089                                   P1=H0XH+P1L          MASK
2E38    5ECB    CSTS 090                                   P0=H0H                 EXECUTION
2E3A    1CC5    CSTS 091                                   P0=P0*-KC0
2E3C    FC40    CSTS 092                  094  OP04A      BR IF P03=0
2E3E    3C45    CSTS 093                                   P0=P0$K40
2E40    7CC2    CSTS 094    OP04A                          STH P DA,B8
2E42    8185    CSTS 095              CICY 005  ISTART N   N=S BITS67
                CSTS 096    *
                CSTS 097    *             SUPERVISOR CALL
3364    55B9    CSTS 098    RROP0A                         T1=G1            SUPERVISOR
3366    2125    CSTS 104                                   U1=0$K20           CALL
3368    2F07    CSTS 106    PICKUP                         H1=0                 EXECUTION
336A    217B    CSTS 107                                   U1=U1+K07
336C    8A02    CSTS 108              BPSW 006  ENTRY      BR
                CSTS 109    *                              *
                CSTS 110    *                              * DIRECT CONTROL OPS
0898    83B4    CSTS 112    RSOP84    CICY 077  OPERR      BR               NO DIRECT CONTROL
0A5E    83B4    CSTS 113    RSOP85    CICY 077  OPERR      BR               NO DIRECT CONTROL
                                                           **********************************
                                                           * CROSS REFERENCE FOR CSECT CSTS *
                                                           **********************************

CSTS 003    CICY 219
CSTS 009    CSTS 006  CSTS 050
CSTS 011    CICY 217
CSTS 019    CICY 220
CSTS 021    CSTS 004  CSTS 012  CSTS 044
CSTS 034    CSTS 019
CSTS 044    CICY 141  CICY 142
CSTS 058    CSTS 064
CSTS 060    CSTS 053
CSTS 071    CSTS 068  DCLL 015  DCLL 041
CSTS 076    CICY 228
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT CSTS *
                                  **********************************

CSTS 083     CSTS 080
CSTS 087     CICY 137
CSTS 094     CSTS 092
CSTS 098     CICY 143
CSTS 106     BMCK 019
CSTS 112     CICY 221
CSTS 113     CICY 222
```

CTRT -- TRANSLATE, TRANSLATE AND TEST, EDIT, EDIT AND MARK

ENTRIES

SSOPDC -- TRANSLATE, TRANSLATE AND TEST
SSOPDE -- EDIT, EDIT AND MARK

OBJECTIVES

TRANSLATE

THE BYTES OF THE 1ST OPERAND ARE USED AS ARGUMENTS TO REFERENCE THE LIST DESIGNATED BY THE 2ND OPERAND ADDRESS. EACH FUNCTION BYTE SELECTED FROM THE LIST REPLACES THE CORRESPONDING ARGUMENT IN THE 1ST OPERAND.
THE BYTES OF THE 1ST OPERAND ARE SELECTED ONE BY ONE FOR TRANSLATION, PROCEEDING LEFT TO RIGHT, UNTIL THE 1ST OPERAND FIELD IS EXHAUSTED.

TRANSLATE AND TEST

THE BYTES OF THE 1ST OPERAND ARE USED AS ARGUMENTS TO REFERENCE THE LIST DESIGNATED BY THE 2ND OPERAND ADDRESS. EACH FUNCTION BYTE THUS SELECTED FROM THE LIST DETERMINES THE CONTINUATION OF THE OPERATION. WHEN THE FUNCTION BYTE IS ZERO, THE OPERATION PROCEEDS BY FETCHING AND TRANSLATING THE NEXT ARGUMENT BYTE. WHEN THE FUNCTION BYTE IS NONZERO, THE OPERATION IS COMPLETED BY INSERTING THE RELATED ARGUMENT ADDRESS IN THE LOW-ORDER 24 BITS OF GENERAL REGISTER 1, AND BY INSERTING THE FUNCTION BYTE IN THE LOW-ORDER BYTE OF GENERAL REGISTER 2.

EDIT

THE FORMAT OF THE SOURCE, 2ND OPERAND, IS CHANGED FROM PACKED TO ZONED AND IS MODIFIED UNDER CONTROL OF THE PATTERN IN THE 2ND OPERAND.
DURING THE EDITING PROCESS, EACH CHARACTER OF THE PATTERN IS AFFECTED AS FOLLOWS.
1. IT IS LEFT UNCHANGED.
2. IT IS REPLACED BY THE 1ST CHARACTER IN THE PATTERN CALLED THE FILL CHARACTER.
3. IT IS REPLACED BY A SOURCE DIGIT EXPANDED TO ZONED-FORMAT.

WHICH ACTION TAKES PLACE IS DETERMINED BY -- THE TYPE OF PATTERN CHARACTER, THE SIGNIFICANCE INDICATOR, AND WHETHER THE SOURCE DIGIT IS ZERO.

PATTERN CHARACTERS

0010 0000	DIGIT SELECTOR. FILL CHARACTER OR ZONED DIGIT REPLACES THE SOURCE DIGIT DEPENDING ON SOURCE DIGIT AND SIGNIFICANCE INDICATOR.
0010 0001	SIGNIFICANCE STARTER. SAME AS DIGIT SELECTOR, EXCEPT 0 CAN START SIGNIFICANCE.
0010 0010	FIELD SEPARATOR. FILL CHARACTER REPLACES SOURCE DIGIT, AND SIGNIFICANCE INDICATOR IS TURNED OFF.
OTHER --	MESSAGE CHARACTER. THESE CHARACTERS REMAIN UNCHANGED OR ARE REPLACED BY THE FILL CHARACTER DEPENDING ON SIGNIFICANCE INDICATOR.

EDIT AND MARK

SAME AS EDIT. IN ADDITION, THE ADDRESS OF THE 1ST SIGNIFICANT RESULT DIGIT IS INSERTED IN BITS 8-31 OF GENERAL REGISTER 1.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              CTRT 001    T              TRANSLATE  AND TRANSLATE AND TEST          C.V.PERKINS
              CTRT 002    *
2DD0    AB25  CTRT 003     SSOPDC    CICY 056  ADDERR N   BR IF H1=NZ          BR IF INVALID ADDER
2DD2    2607  CTRT 004                                    D0=0
2DD4    2E07  CTRT 005     NXTCHR                          H0=0
2DD6    5F10  CTRT 006                                    RDB H1 U           READ ARGUMENT BYTE
2DD8    6F3B  CTRT 007                                    H1C=H1+V1            CALCULATE
2DDA    6E2D  CTRT 008                                    H0C=H0+V0+C          SECOND
2DDC    667D  CTRT 009                                    D0C=D0+D1+C              OPERAND
2DDE    A225  CTRT 010            CICY 056  ADDERR N   BR IF D0=NZ          ADDERR ERROR IF BRANCH
2DE0    5FFC  CTRT 011                                    RDB H1 H           READ TABLE
2DE2    F06D  CTRT 012                 017  TRT          BR IF G07=1          BR IF TRT INSTR
2DE4    7F18  CTRT 013                                    STB H1 U+1
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS
2DE6  25FF    CTRT 014                                    G1=G1+KFF            DECREMENT COUNT
2DE8  F4D5    CTRT 015                  005     NXTCHR     BR IF AC=1
2DEA  8185    CTRT 016                CICY 005  ISTART N   N=S BITS67
2DEC  AB01    CTRT 017    TRT           022     NONZRO N   BR IF H1=NZ
2DEE  5004    CTRT 018                                    U=U+1
2DF0  25FF    CTRT 019                                    G1=G1+KFF
2DF2  F4D5    CTRT 020                  005     NXTCHR     BR IF AC=1
2DF4  AA74    CTRT 021                CICY 003  CHECKX     BR
2800  2B33    CTRT 022    NONZRO 0                        T1=0$K03
2802  2B2D    CTRT 023                                    T1=T1+K20
2804  7FA0    CTRT 024                                    STB H1 AS,T
2806  2BFD    CTRT 025                                    T1=T1+KF0
2808  70AA    CTRT 026                                    STH U AS,T-2
280A  2707    CTRT 027                                    D1=0
280C  77AC    CTRT 028                                    STB D1 AS,T
280E  25FF    CTRT 029                                    G1=G1+KFF
2810  F495    CTRT 030                  032     CDE1       BR IF AC=1
2812  A824    CTRT 031                CLOG 063  OP95A      BR
2814  A820    CTRT 032    CDE1        CLOG 061  OP95C   0  BR
2A78  AB25    CTRT 033    SSOPDE      CICY 056  ADDERR N   BR IF H1=NZ
2A7A  14A3    CTRT 034                                    G0=G0*-K0A           RST G04 AND G06
2A7C  5569    CTRT 035                                    D0=G1                MOVE LENGTH TO D0
2A7E  25F5    CTRT 036                                    G1=0$KF0             PUT EBCDIC ZONE IN G1
2A80  C904    CTRT 037                  039     EBCDIC     BR IF P1 BIT4=0      TEST FOR NO ASCII
2A82  2555    CTRT 038                                    G1=0$K50             PUT IN ASCII ZONE
2A84  5B1C    CTRT 039    EBCDIC                          RDB T1 U+0           READ FIRST PATTERN
2A86  5BF9    CTRT 040                                    H1=T1                T1 HAS FILL CHAR
2A88  0F2D    CTRT 041    TSTPAT                          Z=H1<K20             TEST FOR PATTERN CHAR
2A8A  E096    CTRT 042                  048     NOPAT      BR IF HZNZ           BRANCH IF NOT
2A8C  F0B9    CTRT 043                  065     DIGSEL     BR IF LZ=0           BR TO DIGIT SELECT
2A8E  0F1B    CTRT 044                                    Z=H1<K01             TEST FOR SIG START
2A90  F0B7    CTRT 045                  064     SIGST      BR IF LZ=0
2A92  0F2B    CTRT 046                                    Z=H1<K02             TEST FOR FIELD SEP
2A94  F0B1    CTRT 047                  055     FLDSEP     BR IF LZ=0
2A96  E01B    CTRT 048    NOPAT         050     NOFILL     BR IF G0 BIT6=1      TEST S STAT
2A98  7B10    CTRT 049    FILL                            STB T1 U+0           STORE FILL
2A9A  5004    CTRT 050    NOFILL                          U=U+1                UPDATE ADDRESS
2A9C  26FF    CTRT 051    NEXT                            D0=D0+KFF            DECREMENT LENGTH
2A9E  F4A4    CTRT 052                  058     FINISH     BR IF AC=0           TEST FOR END
2AA0  5F10    CTRT 053                                    RDB H1 U+0           READ NEXT PATTERN CHR
2AA2  AA88    CTRT 054                  041     TSTPAT     BR
2AB0  1427    CTRT 055    FLDSEP                          G0=G0*-K22           RST G02 AND G06
2AB2  3443    CTRT 056                                    G0=G0$K04            SET G05
2AB4  AA98    CTRT 057                  049     FILL       BR
2AA4  E02D    CTRT 058    FINISH        061     SSCH2      BR IF G0 BIT6=1      TEST S STAT
2AA6  E42B    CTRT 059                  063     SSCOD2     BR IF G0 BIT2=1      TEST FOR ZERO FIELD
2AA8  AA74    CTRT 060    SSCODX      CICY 003  CHECKX     BR
2AAC  E428    CTRT 061    SSCH2         060     SSCODX     BR IF G0 BIT2=0      TEST FOR ZERO FIELD
2AAE  A820    CTRT 062                CLOG 061  OP95C   0  BR
2AAA  A824    CTRT 063    SSCOD2      CLOG 063  OP95A      BR
2AB6  1443    CTRT 064    SIGST                           G0=G0*-K04           RST G05
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
2AB8    C03F    CTRT 065    DIGSEL         068  LODIG      BR IF GO BIT4=1        TEST FOR LO DIGIT
2ABA    A325    CTRT 066                CICY 056  ADDERR N   BR IF D1=NZ
2ABC    5738    CTRT 067                                     RDB D1 V+1             READ NEXT SOURCE CHAR
2ABE    4755    CTRT 068    LODIG                            G1=D1XL$G1H            TRNS PACHED TO ZONED
2AC0    F0E3    CTRT 069                   086  DGTZRO     BR IF LZ=0             SOURCE DIGIT ZERO
2AC2    0769    CTRT 070                                     Z=D1+K60               MASK FOR NON DEC DIGT
2AC4    F4E1    CTRT 071                   085  NONDEC     BR IF AC=1  **BRANCH IF GREATER THAN 9
2AC6    E05B    CTRT 072                   082  ISSSTA     BR IF GO BIT6=1        TEST S STAT
2AC8    F05A    CTRT 073                   082  ISSSTA     BR IF GO BIT7=0        TEST FOR EDT AND MARK
2ACA    7EE2    CTRT 074                                     STH H DA,BC              KEEP H
2ACC    5BE9    CTRT 075                                     H0=T1                  KEEP T1
2ACE    2F07    CTRT 076                                     H1=0
2AD0    2B17    CTRT 077                                     T1=0$K11
2AD2    7FA8    CTRT 078                                     STB H1 AS,T+1
2AD4    70A0    CTRT 079                                     STH U AS,T+0  **SIGNIF DIGIT ADDR IN AUXST
2AD6    5EB9    CTRT 080                                     T1=HC
2AD8    5EE2    CTRT 081                                     RDH H DA,BC
2ADA    3427    CTRT 082    ISSSTA                           G0=G0$K22              SET G02,G06 FIELD NZ
2ADC    7518    CTRT 083    SSTAT                            STB G1 U+1             STORE DIGIT
2ADE    AAEA    CTRT 084                   090  NOSGST     BR
2AE0    9BDC    CTRT 085    NONDEC      CCOM 071  FLGDEX     BR
2AE2    E05D    CTRT 086    DGTZRO         083  SSTAT      BR IF GO BIT6=1        TEST S STAT
2AE4    7B18    CTRT 087                                     STB T1 U+1             STORE FILL
2AE6    D06B    CTRT 088                   090  NOSGST     BR IF GO BIT5=1        TEST FOR SIG START
2AE8    3463    CTRT 089                                     G0=G0$K06              SIG START S ON
2AEA    5773    CTRT 090    NOSGST                           D1=D1XH
2AEC    0769    CTRT 091                                     Z=D1+K60  **MASK FOR NON DECIMAL DIGIT
2AEE    F4F5    CTRT 092                   095  SIGN       BR IF AC=1  **  BRANCH IF GREATER THAN 9
2AF0    148B    CTRT 093                                     G0=G0<K08                CHANGE FF STATE
2AF2    AA9C    CTRT 094                   051  NEXT       BR
2AF4    F77A    CTRT 095    SIGN           098  PLUS       BR IF D1 BIT3=0
2AF6    07FD    CTRT 096                                     Z=D1<KF0
2AF8    C4FC    CTRT 097                   099  MINUS      BR IF ZNZ
2AFA    1423    CTRT 098    PLUS                             G0=G0*-K02             TURN S OFF
2AFC    2707    CTRT 099    MINUS                            D1=0
2AFE    3443    CTRT 100                                     G0=G0$K04
2B00    AA9C    CTRT 101                   051  NEXT       BR
                                           **********************************
                                           * CROSS REFERENCE FOR CSECT CTRT *
                                           **********************************

CTRT 003    CICY 263  CICY 264
CTRT 005    CTRT 015  CTRT 020
CTRT 017    CTRT 012
CTRT 022    CTRT 017
CTRT 032    CTRT 030
CTRT 033    CICY 265  CICY 266
CTRT 039    CTRT 037
CTRT 041    CTRT 054
CTRT 048    CTRT 042
CTRT 049    CTRT 057
CTRT 050    CTRT 048
```

****
* CROSS REFERENCE FOR CSECT CTRT *
****

CTRT 051	CTRT 094	CTRT 101
CTRT 055	CTRT 047	
CTRT 058	CTRT 052	
CTRT 060	CTRT 061	
CTRT 061	CTRT 058	
CTRT 063	CTRT 059	
CTRT 064	CTRT 045	
CTRT 065	CTRT 043	
CTRT 068	CTRT 065	
CTRT 082	CTRT 072	CTRT 073
CTRT 083	CTRT 086	
CTRT 085	CTRT 071	
CTRT 086	CTRT 069	
CTRT 090	CTRT 084	CTRT 088
CTRT 095	CTRT 092	
CTRT 098	CTRT 095	
CTRT 099	CTRT 097	

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
                DCHN 001   T                  CHANNEL 0 INTERRUPT ROUTINE
                DCHN 038   *                                                  THIS IS THE ENTRY POINT FOR
                DCHN 039   *                                                   ALL CHNL 0 I/O INTERRUPTS FROM
                DCHN 040   *                                                    THE BSWI ROUTINE
23F2  56D2      DCHN 041    ENTRY                      RDH D DA,BA            RD OUT CHNL 0 INT BUFFER
23F4  1773      DCHN 053                               D1=D1*-K07             SET UCW ADR TO HWD 0
23F6  57B5      DCHN 059    NTCNTR                     T1=D1XL                T=C0CX
23F8  1B83      DCHN 060                               T1=T1*-K08 ** AUX B4=1403,B5=2540R,B6=2540P,B7=1052
23FA  3BB5      DCHN 061                               T1=T1$KB0              T=00B4,5,6 OR 7
23FC  5BA0      DCHN 063                               RDB T1 AS,T+0 ** T IS POINTED AT --B4,B5,B6,ORB7
                DCHN 064   *                                                    WHICH IS1403,2540R,2540P,1052
                DCHN 065   *                                                     UNIT ADDRESSES IN ORDER ABOVE
23FE  5468      DCHN 073                               RDH G AS,D+2           READ OUT CHNL STA/OP-FLGS
2400  C251      DCHN 074                    112  ZUNSTA  BR IF D04=1            BR IF PCI
2402  B44A      DCHN 075              DPTT  089  RESINB  BAL ** GO RST S7,INT   BUFFER BIT AND SEC.
2404  4052      DCHN 076                               RDH U DC,9A            READ ZERO'S INTO U
2406  4206      DCHN 077                               V=U                    MAKE V=0000
2408  D146      DCHN 078                    107  INAINT  BR IF G1 BIT5=0        BR IF NOT ACTIVE
240A  C110      DCHN 079                    082  NOPCI   BR IF G1 BIT4=0        BR IF NO PCI FLAG
240C  3485      DCHN 080                               G0=G0$K80              SET PCI BIT IN CHANNEL STATUS
240E  1583      DCHN 081                               G1=G1*-K08             TURN PCI FLAG OFF
2410  556A      DCHN 082    NOPCI                      RDB G1 AS,D-1          RD UNIT STA INTO G1
2412  D119      DCHN 083                    086  DEVEND  BR IF G1 BIT5=1        BR IF DEVICE END
2414  C118      DCHN 084                    086  DEVEND  BR IF G1 BIT4=0        BR IF NOT CHANNEL END
2416  3085      DCHN 085                               U0=U0$K80              TURN ON SECONDARY BIT
2418  5666      DCHN 086    DEVEND                     D=D-1                  SET UCW ADR TO HWD 0
241A  7068      DCHN 087                               STH U AS,D+2           STORE 8000 IN UCW HWD 0
241C  7160      DCHN 088                               STB U1 AS,D+0          ZERO OUT UNIT STATUS
241E  176B      DCHN 089    PCISS                      D1=D1<K06              SET D TO UCW HWD 2
2420  5068      DCHN 090                               RDH U AS,D+2           RD UCW HWD 2
2422  5260      DCHN 091                               RDH V AS,D+0           RD UCW HWD 3
2424  021E      DCHN 092                               RST MMSK K=71            RST LEVEL 1 TRAP LOCKOUT
2426  C080      DCHN 093    CSWST                      RST S4                 RST DE. ALONE FLAG
2428  55E9      DCHN 094    DEALON                     H0=G1                  MOVE UNIT STATUS
242A  5F59      DCHN 095                               G1=H1                  SAVE CPU ADDR BYTE
242C  54F9      DCHN 096                               H1=G0                  MOVE CHANNEL STATUS
242E  5B49      DCHN 097                               G0=T1                  SAVE UNIT ADDRESS
2430  D235      DCHN 098                        100  COMTIO  BR IF D05=1            BR IF COMM. TEST I/O
2432  3000      DCHN 099                               SET S0                 SET RETURN FLAG
2434  2400      DCHN 100    COMTIO                     SET MODE K=00          SET CPU MODE
2436  C1C3      DCHN 101                        120  STAT    BR IF S4=1             BR IF DE. ALONE
2438  A00E      DCHN 102                CCOM    110  FULCSW  BAL                    GO STORE FULL CSW
243A  54B9      DCHN 103    FINISH                     T1=G0                  RESTORE UNIT ADDRESS
243C  55F9      DCHN 104    FRMFIL                     H1=G1                  RESTORE CPU ADDRESS BYTE
243E  2135      DCHN 105    USECOM                     U1=0$K30               STRT TO BUILD ADR OF I/O OLD PSW
2440  8A00      DCHN 106                BPSW    005  ENTRYB  BR                     BR TO STORE OLD PSW
2446  5560      DCHN 107    INAINT                     RDB G1 AS,D+0          RD OUT UNIT STATUS FROM UCW
2448  716A      DCHN 108                               STB U1 AS,D-1          CLEAR UNIT STATUS
244A  7060      DCHN 109                               STH U AS,D+0           STORE 0000 IN UCW HWD 0
244C  2080      DCHN 110                               SET S4                 REMEMBER DE. ALONE
244E  A428      DCHN 111                        094  DEALON  BR                     GO MOVE STATUS FOR STATUS-0,SCSW
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

2450    5666    DCHN 112    ZUNSTA                         D=D-1              DECR D TO UCW BYTE 1
2452    0010    DCHN 113                                   RST S7              RST CHAN-0 INT
2454    1583    DCHN 114                                   G1=G1*-K08         RST PCI FLAG
2456    1413    DCHN 115                                   G0=G0*-K01          RST INT BUFFER BIT
2458    7468    DCHN 116                                   STH G AS,D+2       STORE UPDATED CH STA/OP-FLGS
245A    2485    DCHN 117                                   G0=0$K80           SET PCI ALONE
245C    2505    DCHN 118                                   G1=0$K00           SET 00 UNIT STATUS
245E    A41E    DCHN 119                 089   PCISS       BR
2442    A01E    DCHN 120    STAT     CCOM 123   STATOS     BAL                GO STORE STATUS AND ZERO CSW
2444    A43A    DCHN 121                 103   FINISH      BR                 GO TAKE INTERRUPT
19F6    2400    DCHN 122    MODRST                         SET MODE K=00
19F8    2A07    DCHN 123    CICRTN                         TO=0
19FA    C3FE    DCHN 124                 126   NOTIPL      BR IF BA BIT4=0    BR IF NOT IPL
19FC    A7D6    DCHN 125                DCLL 025   IPLSTP      BR                 GO TO IPL STOP
19FE    AA76    DCHN 126    NOTIPL   CICY 004   CHECK       BR
158C    2400    DCHN 127    CC3SET                         SET MODE K=00
158E    2C75    DCHN 128    CC3                            PO=0$K70
15C0    99F8    DCHN 129                 123   CICRTN      BR
                                                ************************************
                                                * CROSS REFERENCE FOR CSECT DCHN *
                                                ************************************

DCHN 041     BSWI 151
DCHN 082     DCHN 079
DCHN 086     DCHN 083  DCHN 084
DCHN 089     DCHN 119
DCHN 094     DCHN 111
DCHN 100     DCHN 098
DCHN 103     DCHN 121
DCHN 104     FINT 044
DCHN 105     DCLT 045
DCHN 107     DCHN 078
DCHN 112     DCHN 074
DCHN 120     DCHN 101
DCHN 122     DCLC 208
DCHN 123     DCHN 129
DCHN 126     DCHN 124
DCHN 127     DCLA 133  DCLA 173
DCHN 128     DCLA 047
```

DCLA -- I/O INSTRUCTIONS ROUTINE

ENTRY POINTS

IOINST -- MAJOR ENTRY POINT FOR DCLA. ENTRY IS FROM I-CYCLES AFTER ONE OF THE FOLLOWING I/O INSTRUCTIONS IS DECODED -- START I/O, TEST I/O, HALT I/O, OR TEST CHANNEL.

UADRID -- FROM IPL ROUTINE. IPL IS A SIMULATED START I/O, AND USES -DCLA- FOR THE SAME PURPOSE, I.E. , TO GET THE UNIT ADDRESS.

DYPADD , HOHO , STORO -- FROM PR-KB OR ICA ROUTINES. THESE ENTRIES ARE USED TO SHARE THE TESTING OF DEVICE FLAGS WHEN A PR-KB OR COMMUNICATIONS HALT I/O INSTRUCTION IS IN PROCESS.

DESCRIPTION

I/O INSTRUCTIONS ARE DECODED IN THE I-CYCLES ROUTINE, -CICY- , CAUSING A BRANCH TO -DCLA- .
IN -DCLA- , A CHECK IS MADE TO ENSURE THAT PRIVELEGED OPS ARE PERMISSABLE, I.E. , SUPERVISOR STATE. THE CONDITION CODE IS INITIALIZED TO 0 -- THIS IS CHANGED LATER IF THE I/O UNIT IS BUSY. FOR START I/O ONLY, THE CAW IS FETCHED AND CHECKED. FOR ALL I/O INSTRUCTIONS, THE CHANNEL IS IDENTIFIED. WHEN CHANNEL 1 IS IDENTIFIED, THE ROUTINE BREAKS OUT TO EITHER THE FILE OR BURST CHANNEL ROUTINES IF THESE FEATURES ARE PRESENT.
WHEN CHANNEL 0 IS IDENTIFIED, THE ACTIVE BIT IS SET ON IN THE REGISTER USED TO HOLD THE NEW FLAGS/OP BYTE. THE UCW ADDRESS IS GENERATED, THEN THE MICROPROGRAM BRANCHES TO ANOTHER ROUTINE ON THE BASIS OF THE OP CODE, DEVICE TYPE, AND ACTIVITY STATUS.

OBJECTIVES

1. FOR START I/O, FETCH AND CHECK CAW.
2. GENERATE UCW ADDRESS.
3. EXIT EITHER TO DO THE OPERATION REQUIRED, OR TO RETURN TO I-CYCLES AFTER SETTING THE CONDITION CODE.

ADDR	WORD	SEQUENCE NO.	LABEL		NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLA 001	T				I/O INSTRUCTIONS	H.E.BERKEBILE
0886	10CE	DCLA 002	IOINST				RST S K=FC	RST S REG BITS 0 THRU 5
0888	F90C	DCLA 003			005	SUPVSR	BR IF P17=0	BR IF SUPERVISOR MODE
088A	ADFC	DCLA 004			CICY 079	PRIVOP	BR	
088C	2C07	DCLA 005	SUPVSR				P0=0	SET COND. CODE 0
088E	2B45	DCLA 006					T1=0$K40	
0890	3B83	DCLA 007					T1=T1$K08	SET UP CAW ADDRESS
0892	46A6	DCLA 008					D=T	D=0048 ---START OF UCW ADDR
0894	17FD	DCLA 009					D1=D1<KF0	D=00B8-------
0896	804F	DCLA 010			011	IOOP N	N=G0 BITS67	BR ON OP
0F40	8938	DCLA 011	IOOP	0	069	STRTIO	BR	START I/O
0F42	894E	DCLA 012	IOOP	1	095	UADRID	BR	TEST I/O
0F44	894E	DCLA 013	IOOP	2	095	UADRID	BR	HALT I/O
0F46	5009	DCLA 014	IOOP	3			U0=U0	TEST CHANNEL
0F48	C4D0	DCLA 018			044	NOTCH0	BR IF ZNZ	BR IF NOT CHNL 0
0F4A	C7CF	DCLA 020	CHNL0T		042	CC1T	BR IF BA BIT0=1	BR IF CHNL 0 IB IS ON
0F4C	81CC	DCLA 041	CC0T		CCOM 155	CC0B	BR	SET CC=0
0F4E	AED0	DCLA 042	CC1T		CCOM 158	CC1B	BR	SET CC=1
0F50	001B	DCLA 044	NOTCH0				Z=U0<K01	
0F52	C4D7	DCLA 045			049	CHNL1T	BR IF Z=0	BR IF NOT CHNL 1
0F54	95BE	DCLA 047	CC3T		DCHN 128	CC3	BR	BR TO SET CDN CODE 3
0F56	2486	DCLA 049	CHNL1T				SET MODE K=38	SET CHANNEL MODE
0F58	D8CF	DCLA 050			042	CC1T	BR IF GS BIT5=1	BR IF CHAN INT. IS ON
0F5A	2480	DCLA 058					SET MODE K=08	SET FILE MODE
0F5C	F9CF	DCLA 059			042	CC1T	BR IF FFI6=1	BR IF ANY FILE INT. IS ON
0F5E	5632	DCLA 064					RDH D DA,8E	GET STATS
0F60	C64C	DCLA 065			041	CC0T	BR IF D0 BIT0=0	BR IF ACTIVE OFF

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

0F62  AECA    DCLA 066              CCOM 162  CC2B       BR
              DCLA 068  *                 BEGIN READOUT AND CHECK OF CAW FOR START I-O.
0938  5EB8    DCLA 069   STRTIO                          RDH H T+2              READ 1ST HW OF CAW
093A  7EF2    DCLA 071                                   STH H DA,BE            SAVE THE KEY
093C  0EF5    DCLA 072                                   Z=H0*-KF0              MASK LOW 4 BITS OF KEY BYTE
093E  F0C3    DCLA 073                  075  HIADDR     BR IF LZ=0             BR IF LO 4 BITS OF KEY BYTE=0
0940  B042    DCLA 074   PROG    0  DCLR 008  PGESTT     BR                     BAD CAW--PROG CHECK
0942  AB43    DCLA 075   HIADDR          074  PROG    N  BR IF H1=NZ            BR IF HIGH ADDR NON ZERO
0944  52BA    DCLA 081   RDCAW                           RDH V T-2              READ CCW ADDRESS
0946  C249    DCLA 089                                   Z=V0+K40
0948  F4C1    DCLA 091                  074  PROG    0  BR IF AC=1            BR IF CCW ADR OVER MEMORY SIZE
094A  0383    DCLA 092                                   Z=V1*-K08              CHK FOR 8 BOUNDARY
094C  F0C0    DCLA 093                  074  PROG    0  BR IF LZNZ
              DCLA 094  *                 IDENTIFY THE CHANNEL FOR START I-O, TEST I-O, AND HALT I-O.
094E  5009    DCLA 095   UADRID                          U0=U0
0950  C4D4    DCLA 096                  099  NCHNLO     BR IF ZNZ              BR IF NOT CHNL 0
0952  8A7A    DCLA 097                  141  CHNLO      BR
0954  001B    DCLA 099   NCHNLO                          Z=U0<K01
0956  C4E8    DCLA 100                  133  CC3S       BR IF ZNZ              BR IF NOT CHNL 1
0958  5632    DCLA 105   CHLFIL                         RDH D DA,8E            RD STATS INTO D
095A  2BA1    DCLA 107                                   T1=0-K0A               T=00F5
095C  5FAA    DCLA 108                                   RDB H1 AS,T-1          RD OUT CONTROL UNIT ADR OF FILE
095E  5FE1    DCLA 109                                   H0=H1X                 MOVE CROSSED CU ADDR TO BACK-UP
0960  6F11    DCLA 110                                   H1=H1<U1               COMPARE FIRST CU ADDRESS
0962  E0E7    DCLA 111                  125  FILE       BR IF HZ=0             BR IF CU ADDR MATCH
0964  8011    DCLA 120                  121  BRSTOP  N  N=GO BITS67            BURST CHNL--BR ON OP BYTE
0880  AFF2    DCLA 121   BRSTOP 0  DCLB 004  CCWBUR     BR                     START I/O-BR TO CCW FETCH RTNE
0882  9C28    DCLA 122   BRSTOP 1  DCLT 003  TESTIO     BR                     TEST I/O
0884  A25E    DCLA 123   BRSTOP 2  DCLH 004  ENTRY      BR                     HALT I/O
0966  847C    DCLA 125   FILE      FILE 008  ENTRY      BR                     BR TO 2311 ROUTINE
0968  95BC    DCLA 133   CC3S      DCHN 127  CC3SET     BR
              DCLA 134  *                 BEGIN CONSTRUCTION OF UCW ADDRESS IN D0/D1.  FOR NATIVE PRINTER, 00C8.
              DCLA 135  *                 FOR NATIVE READER, 00D8.  FOR NATIVE PUNCH, 00E8.  FOR NATIVE TYPE-
              DCLA 136  *                 WRITER, 00F8.  FOR NATIVE COMMUNICATIONS, 80XX.  FOR NON-NATIVE CHAN-
              DCLA 137  *                 NEL, 20XX.  XX IS DERIVED FROM UNIT ADDRESS, THUSLY--FOR NATIVE COMM-
              DCLA 138  *                 UNICATIONS, XX=U3,U4,U5,U6,U7,0,0,0.  FOR OTHER (NON-NATIVE) SINGLE
              DCLA 139  *                 SUBCHANNELS, XX=U4,U5,U6,U7,U3,0,0,0.  FOR SHARED SUBCHANNELS, XX=
              DCLA 140  *                 0,U1,U2,U3,0,0,0,0.
0A7A  2543    DCLA 141   CHNLO                                G1=0$K04           SET ACTIVE BIT
0A7C  4E66    DCLA 142                                         H=D                H=00B8
0A7E  1FCB    DCLA 143                                         H1=H1<K0C          H=00B4
0A80  271D    DCLA 144   RDNADR                              D1=D1+K10          D=00C8----1ST UCW ADDRESSS
0A82  5BE8    DCLA 145                                         RDB T1 AS,H+1 ** AUX-B4=PRINT,READER,PUNCH,1052
0A84  6B11    DCLA 146                                         T1=T1<U1           CHECK FOR
0A86  C48D    DCLA 147                          251  TESACT     BR IF Z=0             MATCH
0A88  DB01    DCLA 148                          144  RDNADR     BR IF H15=1        BR IF NOT FINISHED
              DCLA 171  *                 AT THIS POINT, EITHER COMMUNICATIONS OR NON-NATIVE.
0A8A  95BC    DCLA 173   NOTIPL     DCHN 127  CC3SET     BR
035C  8A8C    DCLA 228   OPERNL              251  TESACT     BR
              DCLA 246  *                 FORMATION OF THE UCW ADDRESS IS COMPLETE.  NEXT, BREAK OUT TO OTHER
              DCLA 247  *                 ROUTINES ON THE BASIS OF OP CODE, DEVICE TYPE, ACTIVITY STATUS, ETC.
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
0A8C   5E60    DCLA 251    TESACT                         RDH H AS,D+0           RD UCW FIRST HW
0A8E   E024    DCLA 253                  305   NOHALB     BR IF G06=0            BR IF NOT HALT I/O
0A90   8F7A    DCLA 274    HALTRP  ERRQ 082   RDCHST     BAL    ** GO GET CHAN. AND UNIT STATUS IN G,H
0A92   07FD    DCLA 275                                   Z=D1<KF0               LOOK FOR 1052
0A94   E0CC    DCLA 276                  278   DYPADD     BR IF HZNZ             BR IF OTHER NATIVE
0A96   A46C    DCLA 277            DYPE 221   HLTTW      BR
0AC0   D518    DCLA 278    DYPADD        282   VANSOI     BR IF G11=0              BRANCH IF NOT CHAINED
0AC2   15C5    DCLA 279    HOHO                           G1=G1*-KC0             RST CDA AND CC FLAGS
0AC4   7460    DCLA 280                                   STH G AS,D+0            STORE UPDATED CH STAT IN UCW
0AC6   8A9E    DCLA 281                  285   STORO      BR
0A98   C41E    DCLA 282    VANSOI        285   STORO      BR IF G00=0            BR IF NO SECONDARY
0A9A   D123    DCLA 283                  287   GOBACK     BR IF G15=1            BR IF ACTIVE
0A9C   DA23    DCLA 284                  287   GOBACK     BR IF H05=1            BR IF DEVICE END
0A9E   4E52    DCLA 285    STORO                          RDH H DC,9A            READ ZERO CONSTANT INTO H
0AA0   A030    DCLA 286            CCOM 135   STATUS     BR                     STORE STATUS CSW
0AA2   81CC    DCLA 287    GOBACK  CCOM 155   CC0B       BR                     SET CC0
0AA4   F037    DCLA 305    NOHALB        354   TIOBR      BR IF G07=1            BR IF TEST I/O
0AA6   DB35    DCLA 306                  369   CC2        BR IF H1 BIT5=1        BR ON ACTIVE BIT
0AA8   CE3F    DCLA 320                  368   NOTTES     BR IF H00=1            BR IF SEC. ON
0AAA   5669    DCLA 324    CCHCCW                         D0=D0
0AAC   E0B1    DCLA 325                  327   DCLBBR     BR IF HZ=0             BR IF NATIVE
0AAE   81F8    DCLA 326                  402   BURKEY     BAL                    GO SET UP THE KEY
0AB0   3773    DCLA 327    DCLBBR                         D1=D1$KC7              SET D TO UCW HWD 0
0AB2   95F4    DCLA 329            DCLB 081   RDHW1      BR
0AB6   DB3C    DCLA 354    TIOBR         367   SETIT      BR IF H15=0 ** BR IF NAT.IS NOT ACTIVE
0AB8   CE34    DCLA 365                  369   CC2        BR IF H00=0            SET BUSY IF NO CHAN.END
0ABA   DF35    DCLA 366                  369   CC2        BR IF H11=1            BR IF STILL CHAINED--SET BUSY
0ABC   3613    DCLA 367    SETIT                          D0=D0$K01              REMEMBER TEST IO FOR NATIVES
0ABE   8765    DCLA 368    NOTTES        392   INASIO N   N=D1 BITS23 ** GO TO NAT.TEST I/O---START-SEC. ON
0AB4   AECA    DCLA 369    CC2     CCOM 162   CC2B       BR                     SET CC=2
               DCLA 390    *                BREAKOUT BY DEVICE, FOR START I-O, ACTIVE=0, SECONDARY=1.
0A60   8CC8    DCLA 392    INASIO 0  DPTT 006   TEST00     BR                   PR,SEC BIT=1 START I/O
0A62   9F24    DCLA 394    INASIO 1  ERRQ 004   REDUC      BR                   RD,SEC BIT=1,START IO OR TIO
0A64   9F24    DCLA 395    INASIO 2  ERRQ 004   REDUC      BR                   PU,SEC BIT=1,START IO OR TIO
0A66   B31C    DCLA 400    INASIO 3  DYPE 202   TWSEC7     BR                   TW,SEC BIT=1,START IO
01F8   52FF    DCLA 402    BURKEY                         H1=STP0                H1=CURRENT CPU KEY
01FA   526C    DCLA 403                                   ISK STP0 AS,D+2        READ OUT CHNL/COMM KEYS
01FC   52EF    DCLA 404                                   H0=STP0                H0= CHNL/COMM KEYS
01FE   51F2    DCLA 405                                   RDB U1 DA,BE           READ OUT NEW KEY
0200   C606    DCLA 410                  422   MPXKEY     BR IF D00=0            BR IF MPX CHNL
0202   41E5    DCLA 412                                   H0=U1XL$H0H            OR OLD MPX AND NEW COMM KEYS
0204   C6C9    DCLA 416                  423   KEYOR      BR IF D00=1            UNC BR
0206   41EB    DCLA 422    MPXKEY                         H0=U1H+H0L             OR OLD COMM AND NEW MPX KEYS
0208   42EF    DCLA 423    KEYOR                          STP0=H0                MOVE UPDATED KEY TO STP0
020A   536C    DCLA 424                                   SSK STP1 AS,D+1        STORE UPDATED KEY IN STP1
020C   42FF    DCLA 425                                   STP0=H1                RESTORE CPU KEY
020E   128E    DCLA 426                                   RTN
```

* CROSS REFERENCE FOR CSECT DCLA *

DCLA 002	CICY 247	CICY 248	CICY 249	CICY 250
DCLA 005	DCLA 003			
DCLA 011	DCLA 010			
DCLA 041	DCLA 065			
DCLA 042	DCLA 020	DCLA 050	DCLA 059	
DCLA 044	DCLA 018			
DCLA 049	DCLA 045			
DCLA 069	DCLA 011			
DCLA 074	DCLA 075	DCLA 091	DCLA 093	
DCLA 075	DCLA 073			
DCLA 095	DCLA 012	DCLA 013	DCLL 018	
DCLA 099	DCLA 096			
DCLA 121	DCLA 120			
DCLA 125	DCLA 111			
DCLA 133	DCLA 100			
DCLA 141	DCLA 097			
DCLA 144	DCLA 148			
DCLA 251	DCLA 147	DCLA 228		
DCLA 278	DCLA 276	DYPE 236		
DCLA 279	DYPE 231			
DCLA 282	DCLA 278			
DCLA 285	DCLA 281	DCLA 282	DYPE 227	
DCLA 287	DCLA 283	DCLA 284		
DCLA 305	DCLA 253			
DCLA 327	DCLA 325			
DCLA 354	DCLA 305			
DCLA 367	DCLA 354			
DCLA 368	DCLA 320			
DCLA 369	DCLA 306	DCLA 365	DCLA 366	
DCLA 392	DCLA 368			
DCLA 402	DCLA 326	DCLB 015		
DCLA 422	DCLA 410			
DCLA 423	DCLA 416			

CCW FETCH AND CHECK ROUTINE

THE -DCLB- ROUTINE READS AND CHECKS THE CCW'S FOR IPL, START I/O, COMMAND CHAINING, AND DATA CHAINING. THE ROUTINE IS VERSIONED ACCORDING TO THE CHANNEL FEATURE SELECTED FOR THE CONTROL-STORAGE LOAD, I.E., EITHER CHANNEL 0 -- BYTE MODE, OR CHANNEL 1 -- BURST MODE.

ENTRIES

CCWBUR -- BURST MODE ENTRY FROM START I/O OR IPL. THIS PORTION OF THE ROUTINE SETS THE INITIAL SELECT LATCH AND SETS AN INDICATOR BIT IF THE DEVICE IS BUFFERED.

COMCHN -- COMMUNICATIONS CHAINING ENTRY.

CHAIN -- NATIVE I/O OR BYTE MODE CHAINING ENTRY.

CDACOM -- BYTE MODE DATA CHAINING ENTRY.

ENTRY POINTS CONTINUED

SKIP -- BURST MODE ENTRY TO INCREMENT CCW ADDRESS BY 8 BEFORE CHAINING.

RDHW1X -- BURST MODE COMMAND CHAINING ENTRY.

RDHW1 -- BYTE MODE ENTRY FROM START I/O OR IPL. BURST MODE ENTRY FOR DATA CHAINING WHEN NEXT CCW IS A TIC COMMAND.

NOTIC -- BURST MODE ENTRY FROM LOW-PRIORITY TRAP ROUTINE AFTER DETECTION OF AN INCORRECT TIC ADDRESS. THE CCW ADDRESS IS INCREMENTED IN -DCLB- BEFORE BRANCHING TO THE -DCLR- ERROR ROUTINE.

BSTCDA -- ENTRY FROM -DCLR- ERROR ROUTINE TO PERMIT RETURN FROM TRAP AFTER DATA CHAINING ERROR.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

             DCLB 001    T            CCW FETCH AND CHECK ROUTINE      H.E.BERKEBILE
             DCLB 003    *             BURST MODE ENTRY FROM START I/O
2FF2   C648  DCLB 004     CCWBUR        006  INACT      BR IF D0 BIT0=0      BR IF NOT ACTIVE
2FF4   AECA  DCLB 005    ACTBON    CCOM 162  CC2B       BR                   SET CC=2
2FC8   2045  DCLB 006    INACT                          U0=0$K40
2FCA   7022  DCLB 007                                    STH U DA,8C          STORE UNIT IDENT HWD IN AUX008C
2FCC   2625  DCLB 010                                    D0=0$K20             SET D0 TO BUMP 2
2FCE   2787  DCLB 014                                    D1=0$K88
2FD0   81F8  DCLB 015                 DCLA 402  BURKEY    BAL                  BAL TO STORE KEY
2FD2   2486  DCLB 017                                    SET MODE K=38
2FD4   2880  DCLB 018                                    SET GA K=08          SET INL SELECT LTCH
2FD6   7212  DCLB 019                                    STH V DA,8A          STORE NEXT CCW ADR IN AUX008A
2FD8   24F6  DCLB 020                                    SET MODE K=3F        SET CHNL MODE AND ZONE
2FDA   5622  DCLB 021                                    RDH D DA,8C          RD UNIT IDENT HWD INTO D
2FDC   2407  DCLB 022                                    G0=0                 SET UP BUFFERED
2FDE   2585  DCLB 023                                    G1=0$K80              DEVICE ADDRESSES TABLE
2FE0   3543  DCLB 024                                    G1=G1$K04              ADDRESS - 0084
2FE2   5148  DCLB 025    NOTFIN                         RDB U1 AS,G+1        READ OUT BUFFERED DEVICE ADDRESS
2FE4   6171  DCLB 026                                    U1=U1<D1             LOOK FOR EQUAL ADDRESS
2FE6   C4F7  DCLB 027                      033  MATCH      BR IF Z=0            BR IF EQUAL ADDR
2FE8   C162  DCLB 028                      025  NOTFIN     BR IF G1 BIT4=0      BR IF ADR NOT REACHED 0088 OR 98
2FEA   25CB  DCLB 029                                    G1=G1+K0C            CHANGE ADDR TO 0094 OR 00A4
2FEC   F563  DCLB 030                      025  NOTFIN     BR IF G1 BIT3=1      BR IF ADDR IS 0094
2FEE   2507  DCLB 031                                    G1=0                 HAVE REACHED END AND NO MATCH
2FF0   AFF8  DCLB 032                      034  BSTCC      BR                   CONTINUE WITH CCW FETCH
2FF6   5651  DCLB 033    MATCH                          G1=D0X               G1=04    ACTIVE
2FF8   5212  DCLB 034    BSTCC                          RDH V DA,8A          READ OUT NEXT CCW ADR
2FFA   95F4  DCLB 035                      081  RDHW1      BR
             DCLB 037    *           NATIVE IO AND BYTE MODE CHANNEL CHAIN ENTRY
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

              DCLB 038    *           ENTRY TO FETCH A NEW CCW UPON DATA OR COMMAND CHAINING.  DO/D1 CON-
              DCLB 039    *           TAINS UCW ADDRESS, G1 CONTAINS FLAG AND OP BITS OF PREVIOUS UCW.
15CA    1683  DCLB 043     CHAIN                 D0=D0*-K08
15CC    3613  DCLB 044     CCWCDA                D0=D0$K01          SET "CAME FRM CHAINING" FLAG
15CE    3773  DCLB 045     CDACOM                D1=D1$K07          SET D1 TO UCW BYTE 7
15D0    5260  DCLB 046                           RDH V AS,D+0       READ NEXT CCW ADR
15D2    D25A  DCLB 047                    058  RDHW1X  BR IF DC5=0     BR IF NO COMMU.STAT MOD
15D4    1643  DCLB 048                           D0=D0*-K04         TURN OFF "INCR CCW+8" BIT
15D6    3363  DCLB 049     SKIP                  V1=V1$K06          INCREMENT CCW
15D8    6224  DCLB 050                           V=V+2                 ADR BY 8
15DA    0249  DCLB 058     RDHW1X                Z=V0+K40
15DC    F4C3  DCLB 060                    110  PRGERC  BR IF AC=1      BR IF CCW ADR OVER MEMORY
15DE    5038  DCLB 061                           RDH U V+2          READ FIRST CCW HW
15E0    008B  DCLB 062                           Z=U0<K08
15E2    F0FA  DCLB 063                    084  NOTIC   BR IF LZNZ      BR IF NO TIC
              DCLB 064    *           COMMAND CODE IS TIC.  CHECK AND EXECUTE.
15E4    5119  DCLB 065                           U1=U1
15E6    C4C4  DCLB 066                    111  PRGERB  BR IF ZNZ       BR IF CCW BYTE 1=NZ
15E8    5030  DCLB 067                           RDH U V+0          READ CCW SECOND HW
15EA    0049  DCLB 075                           Z=U0+K40
15EC    F4C5  DCLB 077                    111  PRGERB  BR IF AC=1      BR IF DATA ADR OVER MEMORY
15EE    0183  DCLB 078                           Z=U1*-K08
15F0    F0C4  DCLB 079                    111  PRGERB  BR IF LZNZ      CHK FOR VALID TIC ADR
15F2    4206  DCLB 080                           V=U
15F4    5038  DCLB 081     RDHW1                 RDH U V+2          READ FIRST CCW HW
15F6    008B  DCLB 082                           Z=U0<K08
15F8    F0C5  DCLB 083                    111  PRGERB  BR IF LZ=0      CHK FOR VALID TI
15FA    5119  DCLB 084     NOTIC                 U1=U1
15FC    C4C4  DCLB 085                    111  PRGERB  BR IF ZNZ       BR IF TIC CCW BYTE 1=NZ
15FE    5049  DCLB 086                           G0=U0              MOVE COMMAND TO G0
1600    6224  DCLB 087                           V=V+2              INCREMENT CCW ADR
1602    5038  DCLB 088                           RDH U V+2          READ CCW THIRD HW
1604    0083  DCLB 089                           Z=U0*-K08          CHK FOR VALID FLAG
1606    F094  DCLB 090                    097  PRGERA  BR IF LZNZ      CHK FOR VALID FLAGS
1608    C51F  DCLB 091                    118  CDABR   BR IF G1 BIT0=1 CHK FOR DATA CHAIN
160A    1537  DCLB 092                           G1=G1*-K33         G1=0000X100
160C    F031  DCLB 093     ABCD           114  OUTCMD  BR IF G0 BIT7=1 DECODING OF
160E    E033  DCLB 094                    115  INFCMD  BR IF G0 BIT6=1    THE CCW
1610    C035  DCLB 095                    116  INBCMD  BR IF G0 BIT4=1       COMMAND
1612    D033  DCLB 096                    115  INFCMD  BR IF G0 BIT5=1          BYTE
1614    6224  DCLB 097     PRGERA                V=V+2              INCR V  TO CORRECT NXT CCW ADR
1616    D639  DCLB 099     PRGERR         107  BSTPER  BR IF D0 BIT1=1 BR IF BURST CHNL
1618    3763  DCLB 100                           D1=D1$K06          SET D TO UCW HWD 3
161A    7260  DCLB 104                           STH V AS,D+0       STORE NEXT CCW ADR
161C    B002  DCLB 105     PRGERX  DCLR 002  PRGERR  BR
1638    7212  DCLB 107     BSTPER                STH V DA,8A        STORE UPDATED NXT CCW ADR
163A    B002  DCLB 108             DCLR C02  PRGERR  BR
15C2    6224  DCLB 110     PRGERC                V=V+2              UPDATE V
15C4    6224  DCLB 111     PRGERB                V=V+2                TO CORRECT
15C6    6224  DCLB 112                           V=V+2                   NEXT CCW ADR
15C8    9614  DCLB 113                    097  PRGERA  BR
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS
1630  153B    DCLB 114    OUTCMD                       G1=G1<K03            SET OP BITS TO OUTPUT
1632  151B    DCLB 115    INFCMD                       G1=G1<K01            SET OP BITS TO INP FW
1634  153B    DCLB 116    INBCMD                       G1=G1<K03            SET OP BITS TO INPBKW
1636  9620    DCLB 117                 119  FLAGOR     BR
161E  2405    DCLB 118    CDABR                        G0=0$K00             SET CD INDICATION
1620  15F5    DCLB 119    FLAGOR                       G1=G1*-KF0           CLEAR HIGH FLAGS
1622  6505    DCLB 120                                 G1=G1$U0              OR THE OLD AND NEW FLAGS
1624  5038    DCLB 121    CHKFLG                       RDH U V+2            READ CCW FOURTH HW
1626  4006    DCLB 122                                 U=U
1628  C497    DCLB 123                 099  PRGERR     BR IF Z=0            BR IF CCW COUNT=0000
162A  D63D    DCLB 125                 130  STBURS     BR IF D0 BIT1=1      BR IF BURSTCH
162C  726A    DCLB 127                                 STH V AS,D-2         STORE NEXT CCW ADR
162E  963E    DCLB 128                 132  DECV       BR                   XXXXXXXXXXMAKE ADDRESSABLE
163C  7212    DCLB 130    STBURS                       STH V DA,8A          STORE NEXT CCW ADR
163E  6226    DCLB 132    DECV                         V=V-2                DECREMENT V TO HWD 1
1640  1343    DCLB 133                                 V1=V1*-K04             OF CURRENT CCW
              DCLB 134  *                 LAST TO BE FETCHED IS THE DATA ADDRESS FIELD OF THE NEW CCW.
1642  5230    DCLB 135                                 RDH V V+0            READ CCW SECOND HW
              DCLB 136  *                 CCW IS NOW ALL FETCHED, READY TO BE TURNED OVER TO USING I-O ROUTINE.
              DCLB 137  *                 CCW DATA DISTIBUTED AS FOLLOWS--
              DCLB 138  *                 G0 CONTAINS COMMAND CODE (=0 IF DATA CHAINED TO PRESENT CCW),
              DCLB 139  *                 G1 CONTAINS FLAG/OP BYTE FOR NEW UCW,
              DCLB 140  *                 V0/V1 CONTAINS INITIAL DATA ADDRESS,
              DCLB 141  *                 U0/U1 CONTAINS INITIAL COUNT,
              DCLB 142  *                 D0/D1 CONTAINS UCW ADDRESS+2,
              DCLB 143  *                 D0 BIT 7=1 IF CURRENT CCW IS NOT FIRST IN CURRENT CHAIN, AND
              DCLB 144  *                 NEXT CCW ADDRESS IS IN AUX STORAGE.
1644  D659    DCLB 146                 215  CHCCW      BR IF D01=1           BR IF BURST
1646  C54A    DCLB 148                 174  NOTNAT     BR IF G1 BIT0=0      BR IF NO CDA FLAG
1648  1565    DCLB 149                                 G1=G1*-K60           ZERO OUT CC AND SLI FLAGS
164A  1763    DCLB 174    NOTNAT                       D1=D1*-K06
164C  756A    DCLB 175                                 STB G1 AS,D-1        STORE UCW FLAGS,OP
164E  6666    DCLB 189                                 D=D-2
1650  F265    DCLB 190                 194  CHAKEY     BR IF D07=1          BR IF FROM CHAIN
1652  5FF2    DCLB 191                                 RDB H1 DA,BE         GET SAVED CAW KEY-START I/O
1654  7F60    DCLB 192                                 STB H1 AS,D          STORE KEY IN UCW 00X6
1656  9666    DCLB 193                 195  SETKEY     BR
1664  5F60    DCLB 194    CHAKEY                       RDB H1 AS,D          GET ASSIGNED KEY FROM UCW
1666  6664    DCLB 195    SETKEY                       D=D+2
1668  42FF    DCLB 196                                 STPO=H1              SET KEY FOR NATIVE
166A  8721    DCLB 203                 245  NATSIO N   N=D1 BITS23         BR TO DIF NAT IO STIO
              DCLB 204  *                 IF CHANNEL START I-O, GO ON TO INITIAL SELECTION.
              DCLB 205  *                 IF CHANNEL CMD CHAIN, GO ON TO INITIAL SELECTION, THEN RTN FROM TRAP.
              DCLB 206  *                 IF CHANNEL DATA CHAIN, OR NATIVE CMD OR DATA CHAIN, RETURN FROM TRAP.
1658  A041    DCLB 215    CHCCW            231  NOTCDA N   BR IF G0=NZ          BR IF CMD IS NZ (IF NOT CDA)
165A  C160    DCLB 219                 223  BSTCDA     BR IF G1 BIT4=0      BR IF NO PCI BIT
165C  7602    DCLB 220                                 STH D DA,88          STORE UNIT IDENT IN IB
165E  2D40    DCLB 221                                 SET GB K=04          SET INT BUFFER LATCH
1660  1D00    DCLB 223    BSTCDA                       RST GB K=80          RST CDA REQ LTCH
1662  0244    DCLB 227                                 RTN G MMSK2=0        RETURN FROM TRAP
1040  8DEE    DCLB 231    NOTCDA 0  DCLC 014  ENTRY    BR                   EXIT TO INITIAL SELECTION
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                DCLB 243    *                IF NATIVE START I-O, BRANCH TO INDIVIDUAL ROUTINES.
1020     8BF2   DCLB 245     NATSIO 0  DPTS 043  PRSIO      BR                     PRINTER START I/O SEC BIT=0
1022     9912   DCLB 247     NATSIO 1  ERDR 053  NRDR       BR                     RDR START IO,SEC BIT=0
1024     9912   DCLB 248     NATSIO 2  ERDR 053  NRDR       BR                     PCH START IO,SEC BIT=0
1026     A59A   DCLB 253     NATSIO 3  DYPE 114  TWOFF7     BR                     TW,START IO,SEC BIT=0
                DCLB 257    *                  *********    END OF DCHB (CCW FETCH ROUTINE).    **********
                                                ***********************************
                                                * CROSS REFERENCE FOR CSECT DCLB *
                                                ***********************************

DCLB 004     DCLA 121
DCLB 006     DCLB 004
DCLB 025     DCLB 028  DCLB 030
DCLB 033     DCLB 027
DCLB 034     DCLB 032
DCLB 043     DPTR 074  ECOL 070
DCLB 049     DCLC 164
DCLB 058     DCLB 047  DCLC 165
DCLB 081     DCLA 329  DCLB 035  DCLE 025
DCLB 084     DCLB 063  DCLE 028
DCLB 097     DCLB 090  DCLB 113
DCLB 099     DCLB 123
DCLB 107     DCLB 099
DCLB 110     DCLB 060  DCLE 027
DCLB 111     DCLB 066  DCLB 077  DCLB 079  DCLB 083  DCLB 085
DCLB 114     DCLB 093
DCLB 115     DCLB 094  DCLB 096
DCLB 116     DCLB 095
DCLB 118     DCLB 091
DCLB 119     DCLB 117
DCLB 130     DCLB 125
DCLB 132     DCLB 128
DCLB 174     DCLB 148
DCLB 194     DCLB 190
DCLB 195     DCLB 193
DCLB 215     DCLB 146
DCLB 223     DCLB 219  DCLR 066
DCLB 231     DCLB 215
DCLB 245     DCLB 203
```

DCLC DESCRIPTIVE TEXT

CHANNEL INITIAL SELECTION

THIS ROUTINE PROVIDES THE INITIAL SELECTION SEQUENCE FOR DEVICES ATTACHED TO THE STANDARD INTERFACE. THE ROUTINE IS VERSIONED ACCORDING TO THE CHANNEL FEATURE, EITHER BYTE CHANNEL-0, OR BURST CHANNEL-1.

BURST CHANNEL ENTRY POINTS

ENTRY -- INITIAL SELECTION ENTRY, OR RESELECTION ENTRY IF CHAINING, FOR START I/O OR IPL AFTER COMPLETING CCW FETCH IN -DCLB- ROUTINE.

NOPCIB -- ENTRY FOR I/O INTERRUPT OR INACTIVE TEST I/O TO GET DEVICE ON THE INTERFACE.

HLTENT -- INACTIVE HALT I/O ENTRY TO GET DEVICE ON THE INTERFACE.

ADRABC -- FROM CHANNEL LOW-PRIORITY-TRAP STATUS ROUTINE -DCLE- TO CONTINUE TESTING STATUS. FROM THIS POINT, THE ROUTINE CAN EITHER COMMAND CHAIN, TRY TO LOAD THE INTERRUPT BUFFER, OR STORE THE CSW.

ADR11A -- FROM -DCLE- STATUS ROUTINE TO TRY LOADING THE INTERRUPT BUFFER OR TO STORE THE CSW.

TRYIBS -- FROM -DCLE- STATUS ROUTINE TO TRY LOADING THE INTERRUPT BUFFER.

ADR22A -- ENTRY TO DO A TIMEOUT ON FALL OF OPERATIONAL-IN, THEN EITHER RETURN FROM TRAP OR GO TO I-CYCLES.

BYTE CHANNEL ENTRY POINTS

ENTRY -- INITIAL SELECTION ENTRY, OR RESELECTION ENTRY IF CHAINING, FOR START I/O OR IPL AFTER COMPLETING CCW FETCH IN -DCLB- ROUTINE.

BEGIN -- ENTRY TO SELECT THE DEVICE FOR HALT I/O, I/O INTERRUPT, OR WHEN A CHANNEL ERROR IS DETECTED.

NOCHE -- ENTRY FROM LOW-PRIORITY-TRAP STATUS ROUTINE -DCLE- TO EITHER COMMAND CHAIN OR SET UP FOR INTERRUPT.

IPLCHK
STATUS -- ENTRY FROM -DCLE- STATUS ROUTINE TO SET UP FOR INTERRUPT.

CSWFRM -- FROM INTERRUPT ROUTINE OR TEST I/O ROUTINE TO PUT TOGETHER CSW.

ACTPCI -- FROM INTERRUPT ROUTINE TO STORE STATUS IN THE CSW FOR ACTIVE PCI.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              DCLC 001    T              360 CHANNEL INITIAL SELECTION                    H.E.BERKEBILE
              DCLC 003    *                                          COUNT FIELD, HIGHORDERBYTE=U0
              DCLC 004    *                                          COUNT FIELD, LOW ORDER BYTE=U1
              DCLC 005    *                                          DATA ADDRESS, HIGH ORDER BYRE=V0
              DCLC 006    *                                          DATA ADDRESS, LOW ORDER BYTE =V1
              DCLC 007    *                                          (CMD)=G0
              DCLC 008    *                                          FLAGS AND OP BYTE =G1
              DCLC 009    *                                          TIMEOUT COUNTER =D0
              DCLC 010    *                                          DEVICE ADDRESS =D1
              DCLC 011    *                                          G1=0001 0000---I/O INTERRUPT
              DCLC 012    *                                          G1=0000 0000---TEST I/O
              DCLC 013    ASEQ    AL07=6E
0DEE    C176  DCLC 014     ENTRY          018  NOPCIB     BR IF G1 BIT4=0      BR IF NO PCI BIT
0DF0    7602  DCLC 015                                   STH D DA,88          STORE UNIT IDENT IN INT BUFFER
0DF2    2D40  DCLC 016                                   SET GB K=04          SET INT BUFFER LATCH
0DF4    8DF8  DCLC 017                 019  ENTRY2     BR
0DF6    0D40  DCLC 018     NOPCIB                      RST GB K=04              RESET IB LTCH
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0DF8    2B80    DCLC 019    ENTRY2                         SET GA K=08          SET INL SELECT LATCH
0DFA    4F7F    DCLC 020    HLTENT                         GB/OUT=D1          PUT ADR ON BUS OUT
0DFC    2B84    DCLC 021                                   SET GA K=28          SET ADDR-OUT
0DFE    2BC4    DCLC 022                                   SET GA K=2C          SET SEL-O AND ADR-O
0E00    C9DD    DCLC 023    ADR1          069   OPINUP     BR IF GT BIT4=1      BR IF OP IN UP
0E02    FDA1    DCLC 024                  039   CUBUSY     BR IF GT BIT3=1      BR IF STA-IN UP
0E04    262B    DCLC 025                                   DO=DO+K02            INCREMENT COUNTER
0E06    E600    DCLC 026                  023   ADR1       BR IF DO BIT2=0      BR IF NO TIMEOUT
0E08    05C3    DCLC 027    SLIN                           Z=G1*-K0C            MASK G1 BITS6 AND 7
0E0A    F094    DCLC 028                  033   ADR28      BR IF L7NZ           BR IF START I/O
0E0C    F514    DCLC 029                  033   ADR28      BR IF G1 BIT3=0      BR IF TEST I/O
0E0E    2323    DCLC 030                                   V1=0$K02             SET INTFCE CTRL CHECK
0E10    2207    DCLC 031                                   V0=0                 SET CO UNIT STATUS
0E12    8E3A    DCLC 032                  052   ADR33C     BR
0E14    DD9F    DCLC 033    ADR28         038   INFCHK     BR IF GT BIT1=1      BR IF SEL IN DOWN
0E16    2B00    DCLC 034    ADR29                          SET GA K=00          RESET ALL TAGS OUT
0E18    F21F    DCLC 035                  038   INFCHK     BR IF DO BIT7=1      BR IF CHAINED-TO CCW
0E1A    C3CB    DCLC 036                  060   IPL002     BR IF BA BIT4=1      BR IF IPL
0E1C    ADF6    DCLC 037    ADR30   CCOM 166   CC3B       BR
0E1E    A154    DCLC 038    INFCHK  DCLR 155   INFCHK     BR
0E20    5F4F    DCLC 039    CUBUSY                         GO=GB/IN             BUSS IN TO GO
0E22    FC9F    DCLC 040                  038   INFCHK     BR IF GS BIT3=1      BR IF PARITY ERROR
0E24    2B84    DCLC 041                                   SET GA K=28          DROP SEL-OUT,KEEP ADR-O&INL SEL
0E26    E61F    DCLC 042    ADR10         038   INFCHK     BR IF DO BIT2=1      BR IF TIMEOUT
0E28    FDAE    DCLC 043                  046   ADR9       BR IF GT BIT3=0      BR IF STA-IN DOWN
0E2A    262B    DCLC 044                                   DO=DO+K02            INCREMENT COONTER
0E2C    FDA7    DCLC 045                  042   ADR10      BR IF GT BIT3=1      BR IF STA-IN STILL UP
0E2E    2B80    DCLC 046    ADR9                           SET GA K=08          DROP ADR-OUT ,KEEP INL SEL ON
0E30    F21F    DCLC 047                  038   INFCHK     BR IF DO BIT7=1      BR IF CHAINED-TO CCW
0E32    E149    DCLC 048                  059   STRTIO     BR IF G1 BIT6=1      BR IF
0E34    F149    DCLC 049                  059   STRTIO     BR IF G1 BIT7=1        START I/O
0E36    2307    DCLC 050    ADR33                          V1=0                 SET CO CHANNEL STATUS
0E38    5429    DCLC 051    ADR33B                         V0=GO                MOVE UNIT STATUS TO V0
0E3A    72A2    DCLC 052    ADR33C                         STH V DA,AC          SAVE STATUS
0E3C    2B00    DCLC 053                                   SET GA K=00          DROP ALL TAGS OUT
0E3E    2400    DCLC 054    ADR34                          SET MODE K=00
0E40    5EA2    DCLC 055                                   RDH H DA,AC          RESTORE CHANNEL STATUS
0E42    2A07    DCLC 056                                   TO=0
0E44    A01E    DCLC 057              CCOM 123   STATOS     BAL                  BAL  TO RTHE TO STORE CSW
0E46    9C6A    DCLC 058              DCLT 041   INTRTX     BR                   RTN FROM BAL FOR I/O INT ONLY
0E48    C3D8    DCLC 059    STRTIO        067   ADREBR     BR IF BA BIT4=0      BR IF NOT IPL
0E4A    A7D6    DCLC 060    IPL002  DCLL 025   IPLSTP     BR                   GO TO IPL STOP
0E4C    F537    DCLC 061    ADR32         050   ADR33      BR IF G13=1          BR IF I/O INTERRUPT
0E4E    5449    DCLC 062                                   GO=GO
0E50    C486    DCLC 063                  050   ADR33      BR IF ZNZ            BR IF STA NZ
0E52    2B00    DCLC 064                                   SET GA K=00            RESET INL SEL
0E54    810C    DCLC 065    ADR33A  CCOM 155   CCOB       BR                   SET CDN CODE 0 & RTN TO ICYC
0E56    2B84    DCLC 066    HALTIO                         SET GA K=28          RAISE ADR-O AND DROPSEL-O
0E58    2307    DCLC 067    ADREBR                         V1=0                 CHNL STATUS = 0
0E5A    A7B6    DCLC 068                  170   ADRE       BR                   GO SET I/O INTERRUPT
0E5C    2BC0    DCLC 069    OPINUP                         SET GA K=0C            RESET ADR-O
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0E5E    0D10    DCLC 070                                    RST GB K=01          RST SUPR OUT
0E60    E61F    DCLC 071    ADR2          038     INFCHK     BR IF DO BIT2=1    BR IF TIMEOUT
0E62    CDE9    DCLC 072                  075     ADINUP     BR IF GT BIT0=1    BR IF ADR-IN UP
0E64    262B    DCLC 073                                     D0=D0+K02          INCREMENT COONTER
0E66    CDE0    DCLC 074                  071     ADR2       BR IF GT BIT0=0    BR IF ADR-IN NOT UP Y
0E68    4F4F    DCLC 075    ADINUP                           GB/OUT=G0          PUT CMD ON BUSS OUT
0E6A    5F4F    DCLC 076                                     G0=GB/IN           PIT DEVICE ADR IN G0
0E6C    FC9F    DCLC 077                  038     INFCHK     BR IF GS BIT3=1    BR IF PARITY ERROR
0E6E    6471    DCLC 078                                     G0=G0<D1           COMPARE ADDRESS
0E70    C49E    DCLC 079                  038     INFCHK     BR IF ZNZ          BR IFADR MISMATCH
0E72    054B    DCLC 080                                     Z=G1<K04           TEST FOR HALT I/O
0E74    C4D7    DCLC 081                  066     HALTIO     BR IF Z=0          BR IF HLT I/O
0E76    28C2    DCLC 082                                     SET GA K=1C          SET CMD-0
0E78    9E86    DCLC 083                  085     ADR3       BR
                DCLC 084    AEND
1E86    F605    DCLC 085    ADR3          124     INFCKK     BR IF DO BIT2=1    BR IF TIMEOUT
1E88    FD8F    DCLC 086                  089     STINUP     BR IF GT BIT3=1    BR IF STA-IN UP
1E8A    262B    DCLC 087                                     D0=D0+K02          INCREMENT COONTER
1E8C    FD86    DCLC 088                  085     ADR3       BR IF GT BIT3=0    BR IF STA-IN NOT UP YET
1E8E    5F4F    DCLC 089    STINUP                           G0=GB/IN           G0=STATUS ON BUSS-IN
1E90    FC85    DCLC 090                  124     INFCKK     BR IF GS BIT3=1    BR IF PARITY ERROR
1E92    E123    DCLC 091                  099     NOTTST     BR IF G1 BIT6=1    BR IF
1E94    F123    DCLC 092                  099     NOTTST     BR IF G1 BIT7=1      START I/O
1E96    2B88    DCLC 093    TSTIS                            SET GA K=48          SET SRV-OUT,INLSELUP
1E98    F605    DCLC 094    ADR31         124     INFCKK     BR IF DO BIT2=1    BR IF TIMEOUT
1E9A    C9A0    DCLC 095                  098     ADR32B     BR IF GT BIT4=0    BR IF OP-IN DOWN
1E9C    262B    DCLC 096                                     D0=D0+K02          INCREMENT TMOUT CNTR
1E9E    C999    DCLC 097                  094     ADR31      BR IF GT BIT4=1    BR IF OP-IN STILL UP
1EA0    8E4C    DCLC 098    ADR32B        061     ADR32      BR
1EA2    C4B6    DCLC 099    NOTTST        126     NZSTA      BR IF ZNZ          BR IF NZ UNIT STATUS
1EA4    F233    DCLC 100    ZEROST        107     NOBUFF     BR IF D07=1        BR IF CHAINED TO CCW
1EA6    2D04    DCLC 101                                     SET GB K=20        TURN ON BURST LATCH
1EA8    3685    DCLC 102                                     D0=D0$K80          TURN ON BURSTCH/FILE ACTIVE BIT
1EAA    7632    DCLC 103                                     STH D DA,8E        STORE STATS
1EAC    1685    DCLC 104                                     D0=D0*-K80         RESTORE DO TO ORIGINAL VALUE
1EAE    D134    DCLC 105                  108     NOTBUF     BR IF G1 BIT5=0    BR IF NOT BUFFERED DEVICE
1EB0    2D02    DCLC 106                                     SET GB K=10          TN ON BUF DEV LTCH
1EB2    1543    DCLC 107    NOBUFF                           G1=G1*-K04         RESET BUFFERED DEV INDICATOR
1EB4    8442    DCLC 108    NOTBUF        116     FAAE1      BR
                DCLC 110   ATABLE  ADDR=0442                          THE FOLLOWING TWO WORDS HAVE
                DCLC 111   *                                            FIXED ADDRESSES TO ALLOW A
                DCLC 112   *                                            CONSTANT PATCH CARD TO BE ADDED
                DCLC 113   *                                            TO A CSL DECK TO RUN NHW
                DCLC 114   *                                            DIAGNOSTIC FAAE.
0442    2B48    DCLC 116    FAAE1                            SET GA K=44        SET SRV-OUT & KEEP SEL-OUT UP
0444    AF70    DCLC 118                  121     FAAEEX     BR
                DCLC 119   AEND
2F70    1635    DCLC 121    FAAEEX                           D0=D0*-K30         RST 2 HI BITS OF TIMEOUT CNTR
2F72    C3F7    DCLC 122    IPLBR         125     IPL001     BR IF BA BIT4=1    BR IF IPL
2F74    A7AE    DCLC 123                  206     ADR24      BR
1E84    A154    DCLC 124    INFCKK    DCLR 155    INFCHK     BR
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS
2F76    B488    DCLC 125    IPL001    DCLL 019   IPLWTL     BR
1EB6    C04E    DCLC 126    NZSTA          138   NOCHE      BR IF GO BIT4=0    BR IF NO CH.E. BIT
1EB8    D03C    DCLC 127                   129   NODE       BR IF GO BIT5=0    BR IF NO DEVICE END BIT
1EBA    0D40    DCLC 128                                      RST GB K=04        RESET INT BUFFER LATCH
1EBC    F44F    DCLC 129    NODE           138   NOCHE      BR IF GO BIT3=1    BR IF BSY BIT IN STATUS
1EBE    2D04    DCLC 130                                      SET GB K=20            TURN ON BURST LATCH
1EC0    D148    DCLC 131                   135   NTBFX      BR IF G1 BIT5=0    BR IF NOT BUFFERED DEVICE
1EC2    F247    DCLC 132                   134   RSTG15     BR IF D0 BIT7=1    BR IF CHAINED TO CCW
1EC4    2D02    DCLC 133                                      SET GB K=10        SET BUFFERED DEVICE LATCH
1EC6    1543    DCLC 134    RSTG15                            G1=G1*-K04         RESET BUFFERED DEVICE INDICATOR
1EC8    2285    DCLC 135    NTBFX                             V0=0$K80           TURN ON  BURSTCH/FILE ACTIVE BIT
1ECA    2307    DCLC 136                                      V1=0
1ECC    7232    DCLC 137                                      STH V DA,8E        STORE STATS
1ECE    2307    DCLC 138    NOCHE                             V1=0               CHNL STATUS = 0
1ED0    C55F    DCLC 139                   146   CCFLON     BR IF G1 BIT0=1    BR IF CDA FLG ON
1ED2    D560    DCLC 140    ADRABC         147   ADR11      BR IF G1 BIT1=0    BR IF CCFLG =0
1ED4    04C7    DCLC 141                                      Z=G0*-KCC
1ED6    C4E0    DCLC 142                   147   ADR11      BR IF ZNZ          BR IF BAD STATUS
1ED8    C461    DCLC 143                   147   ADR11      BR IF GO BIT0=1    BR IF BAD STATUS
1EDA    D065    DCLC 144                   154   DECHTP     BR IF GO BIT5=1    BR IF DEVICE END
1EDC    C07B    DCLC 145                   149   ADR12      BR IF GO BIT4=1    BR IF CHNL END
1EDE    15C5    DCLC 146    CCFLON                            G1=G1*-KC0         CDA FLG ON - BRK CHAINING
1EE0    0D10    DCLC 147    ADR11                             RST GB K=01        RESET SUPPRESS OUT
1EE2    A782    DCLC 148                   166   ADR11A     BR    ** TO NOT CHAINING OR BAD STATUS RTNE
1EEA    2D10    DCLC 149    ADR12                             SET GB K=01            SET SUPR OUT
1EEC    2B48    DCLC 150                                      SET GA K=44            SET SERVICE OUT
1EEE    2405    DCLC 151                                      G0=0$K00           G0=00 CHNL STATUS
1EF0    1573    DCLC 152                                      G1=G1*-K07         RST CODED OP BITS
1EF2    AF72    DCLC 153                   122   IPLBR      BR
1EE4    2D10    DCLC 154    DECHTP                            SET GB K=01            SET SUPR OUT
1EE6    2B88    DCLC 155                                      SET GA K=48            SET SRV-OUT
1EE8    E605    DCLC 156    ADR15          124   INFCKK     BR IF D0 BIT2=1    BR IF TIMEOUT
1EEA    C9F0    DCLC 157                   160   ADR14      BR IF GT BIT4=0    BR IF OP IN DOWN
1EEC    262B    DCLC 158                                      D0=D0+K02          INCREMENT TMOUT CNTR
1EEE    C9E9    DCLC 159                   156   ADR15      BR IF GT BIT4=1    BR IF OP-IN STILL UP
1EF0    2645    DCLC 160    ADR14                             D0=0$K40           RESET TMOUT CNTR AND SET
1EF2    3613    DCLC 161                                      D0=D0$K01             UP  D0 TO CHAIN
1EF4    5212    DCLC 162                                      RDH V DA,8A        RD OUT NXT CCW ADR
1EF6    D402    DCLC 163                   165   NOINCR     BR IF GO BIT1=0    BR IF NO STATUS MODIFIER
1EF8    95D6    DCLC 164               DCLB 049   SKIP       BR                 GO INCR CCW ADR+8 BEFORE CHAIN
1E82    95DA    DCLC 165    NOINCR     DCLB 058   RDHW1X     BR                 NO CCW INCR NECESSARY-DO CHAIN
2782    C3B2    DCLC 166    ADR11A         168   NTIPL2     BR IF BA BIT4=0    BR IF NOT IPL
2784    A7DC    DCLC 167               DCLL 028   IPLSTB     BR                 BR TO IPL STA RTNE
27B2    F209    DCLC 168    NTIPL2         186   NZSTCC     BR IF D0 BIT7=1    BR IF CHAINED-TO CCW
27B4    2B88    DCLC 169                                      SET GA K=48        SET SERVICE-OUT
27B6    5429    DCLC 170    ADRE                              V0=G0              PUT UNIT STA IN V0
27B8    C13C    DCLC 171                   173   ADR16      BR IF G1 BIT4=0    BR IF PCI FLG OFF
27BA    3385    DCLC 172                                      V1=V1$K80          PUT PCI BIT IN CH STA
27BC    7222    DCLC 173    ADR16                             STH V DA,8C        TEMP STORE STATUS
27BE    2007    DCLC 174                                      U0=0
27C0    7032    DCLC 175                                      STH U DA,8E        AUX008E = 0000 --RST STATS
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
27C2  CD04    DCLC 176                                   RST GB K=20          RESET BURST LATCH
27C4  E607    DCLC 177    ADR18       191    INFCKX     BR IF D0 BIT2=1      BR IF TIMEOUT
27C6  C9CC    DCLC 178                181    ADR17      BR IF GT BIT4=0      BR IF OP-IN DOWN
27C8  262B    DCLC 179                                   D0=D0+K02            INCREMENT TMOUT COUNTER
27CA  C9C5    DCLC 180                177    ADR18      BR IF GT BIT4=1      BR IF OP-IN STILL UP
27CC  0000    DCLC 181    ADR17                          RST S K=00           NO-OP DELAY
27CE  2B00    DCLC 182                                   SET GA K=00          RESET INITIAL SELECT
27D0  2400    DCLC 183                                   SET MODE K=00        SET CPU MODE AND ZONE
27D2  5E22    DCLC 184                                   RDH H DA,8C          RD OUT  UNIT &CHNL STATUS
27D4  A030    DCLC 185                CCOM 135   STATUS   BR                   BR TO STORE CSW ROUTINE
2788  5429    DCLC 186    NZSTCC                         V0=G0                XFER STATUS TO V0
278A  5349    DCLC 187                                   G0=V1                G0 = CHNL STATUS
278C  EC93    DCLC 188                192    ADR20      BR IF GS BIT2=1      BR IF BURST LTCH ON
278E  D892    DCLC 189    TRYIBS      192    ADR20      BR IF GS BIT5=0      BR IF IBLTCH=0
2790  A7A2    DCLC 190                200    ADR22      BR
2786  A154    DCLC 191    INFCKX      DCLR 155  INFCHK  BR
2792  1635    DCLC 192    ADR20                          D0=D0*-K30
2794  7602    DCLC 193                                   STH D DA,88          STORE UNIT IDENT HWD IN IB
2796  3413    DCLC 194                                   G0=G0$K01            TURN ON STATUS RECD B
2798  2D40    DCLC 195                                   SET GB K=04            TURN ON IB LATCH
279A  ECA2    DCLC 196                200    ADR22      BR IF GS BIT2=0      BR IF BRST LTCH OFF
279C  2B08    DCLC 197                                   SET GA K=40            SRV-0---ACCEPT STATUS
279E  3413    DCLC 198                                   G0=G0$K01            TURN ON STA RECD BIT
27A0  A7A6    DCLC 199                202    ADR23      BR
27A2  2B02    DCLC 200    ADR22                          SET GA K=10            CMD-0---QUE STATUS
27A4  1635    DCLC 201    ADR22A                         D0=D0*-K30
27A6  E607    DCLC 202    ADR23       191    INFCKX     BR IF D0 BIT2=1      BR IF TIMEOUT
27A8  C9AE    DCLC 203                206    ADR24      BR IF GT BIT4=0      BR IF OP IN DOWN
27AA  2628    DCLC 204                                   D0=D0+K02            INCREMENT TMOUT COUNTER
27AC  C9A7    DCLC 205                202    ADR23      BR IF GT BIT4=1      BR IF OP-IN STILL UP
27AE  E680    DCLC 206    ADR24       208    ADR25      BR IF MMSK2=0        BR IF NOT IN TRAP
27B0  0244    DCLC 207                                   RTN G MMSK2=0        RETURN FROM TRAP
2780  99F6    DCLC 208    ADR25     DCHN 122  MODRST     BR                   RTN TO I CYCLES
              DCLC 209    *
              DCLC 210    ATABLE  ADDR=0084
0084  0B0C    DCLC 211    C                              XAUX'CB0C0D0E'       BUFFERED DEVICE STANDARD ADDRESS
              DCLC 212    AEND
              DCLC 213    ATABLE  ADDR=0094
0094  0F15    DCLC 214    C                              XAUX'0F1518FF'       BUFFERED DEVICE STANDARD ADDRESS
              DCLC 215    AEND
```

****
* CROSS REFERENCE FOR CSECT DCLC *
****

```
DCLC 014    DCLB 231
DCLC 018    DCLC 014  DCLH 015
DCLC 019    DCLC 017
DCLC 020    DCLH 014
DCLC 023    DCLC 026
DCLC 033    DCLC 028  DCLC 029
DCLC 038    DCLC 033  DCLC 035  DCLC 040  DCLC 042  DCLC 047  DCLC 071  DCLC 077  DCLC 079
DCLC 039    DCLC 024
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT DCLC *
                                  **********************************
DCLC 042     DCLC 045
DCLC 046     DCLC 043
DCLC 050     DCLC 061  DCLC 063
DCLC 052     DCLC 032
DCLC 059     DCLC 048  DCLC 049
DCLC 060     DCLC 036
DCLC 061     DCLC 098
DCLC 066     DCLC 081
DCLC 067     DCLC 059
DCLC 069     DCLC 023
DCLC 071     DCLC 074
DCLC 075     DCLC 072
DCLC 085     DCLC 083  DCLC 088
DCLC 089     DCLC 086
DCLC 094     DCLC 097
DCLC 098     DCLC 095
DCLC 099     DCLC 091  DCLC 092
DCLC 107     DCLC 100
DCLC 108     DCLC 105
DCLC 116     DCLC 108
DCLC 121     DCLC 118
DCLC 122     DCLC 153
DCLC 124     DCLC 085  DCLC 090  DCLC 094  DCLC 156
DCLC 125     DCLC 122
DCLC 126     DCLC 099
DCLC 129     DCLC 127
DCLC 134     DCLC 132
DCLC 135     DCLC 131
DCLC 138     DCLC 126  DCLC 129
DCLC 140     DCLE 075
DCLC 146     DCLC 139
DCLC 147     DCLC 140  DCLC 142  DCLC 143
DCLC 149     DCLC 145
DCLC 154     DCLC 144
DCLC 156     DCLC 159
DCLC 160     DCLC 157
DCLC 165     DCLC 163
DCLC 166     DCLC 148  DCLE 072
DCLC 168     DCLC 166
DCLC 170     DCLC 068
DCLC 173     DCLC 171
DCLC 177     DCLC 180
DCLC 181     DCLC 178
DCLC 186     DCLC 168
DCLC 189     DCLE 057
DCLC 191     DCLC 177  DCLC 202
DCLC 192     DCLC 188  DCLC 189
DCLC 200     DCLC 190  DCLC 196
DCLC 201     DCLR 075
```

****
* CROSS REFERENCE FOR CSECT DCLC *
****

DCLC 202 DCLC 199 DCLC 205
DCLC 206 DCLC 123 DCLC 203
DCLC 208 DCLC 206

DCLD DESCRIPTIVE TEXT

CHANNEL DATA LOOP

THIS ROUTINE PROVIDES THE DATA LOOP FOR EITHER THE BYTE CHANNEL-0 OR BURST CHANNEL-1, DEPENDING ON WHICH CHANNEL OPTION IS LOADED IN CONTROL STORAGE.

BURST CHANNEL ENTRY POINTS

- 0170 -- TRAP ENTRY FOR DATA SERVICE ON NON-BUFFERED DEVICE.
- OPBRCH -- ENTRY FROM LOW-PRIORITY-TRAP ROUTINE -DCLE- FOR BUFFERED DEVICES.
- DECCNT -- FROM -DCLR- ROUTINE AFTER CHANNEL STATUS IS SET FOR PROTECTION CHECK OR STORAGE WRAP CHECK. PURPOSE OF ENTRY IS TO DECREMENT COUNT, THEN RETURN FROM TRAP OR GO TO I-CYCLES.

BYTE CHANNEL ENTRY POINTS

- ENTRY -- COMMON ENTRY INTO A LOOP THAT CHECKS FOR THE RISE OF SERVICE-IN, STATUS-IN, OR THE FALL OF OPERATIONAL-IN. THIS ENTRY INITIALIZES THE TIMEOUT COUNTER.
- LOOP -- SAME AS -ENTRY- EXCEPT TIMEOUT COUNTER IS NOT INITIALIZED.
- RESET -- FROM LOW-PRIORITY-TRAP ROUTINE -DCLE- OR CHANNEL ERROR ROUTINE -DCLR- TO RESET TAGS-OUT, THEN EITHER RETURN FROM TRAP OR GO TO I-CYCLES.
- SERV -- FROM -DCLE- STATUS ROUTINE WHEN THE DEVICE REQUESTS SERVICE.
- DECREM -- FROM -DCLR- CHANNEL ERROR ROUTINE AFTER CHANNEL STATUS IS SET FOR PROTECTION CHECK OR STORAGE WRAP CHECK. PURPOSE OF ENTRY IS TO DECREMENT COUNT, THEN RETURN FROM TRAP.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS
              DCLD 001    T                    CHANNEL DATA LOOP                    H.E.BERKEBILE
              DCLD 009    ATABLE  ADDR=0170
0170    2240  DCLD 010     LABEL                            LINK G MMSKO=1          FIRST WORD OF TRAP
0172    8153  DCLD 011     OPBRCH          013  OP      N   N=G1 BITS67             BR ON OP
              DCLD 012    AEND
01C0    81EC  DCLD 013     OP      0       035  EXPSTA      BR                      COUNT=0 AND STATUS EXPECTED
01C2    81DC  DCLD 014     OP      1       027  OUTPUT      BR                      OUTPUT COMMAND
01C4    F554  DCLD 015     OP      2       023  RDFWD       BR IF G1 BIT3=0         BR IF SKIP FLAG IS 0
01C6    F551  DCLD 016     OP      3       021  SKIP        BR IF G1 BIT3=1         BR IF SKIP FLAG IS 1
01C8    7F3E  DCLD 017                                      STB GB/IN V-1           STORE BUS IN,ADR -1
01CA    5006  DCLD 018     DECCNT                           U=U-1                   COUNT FIELD -1
01CC    C4E3  DCLD 019                      030  CTZERO      BR IF Z=0               BR IF COUNT = 0
01CE    0240  DCLD 020                                      RTN G MMSKO=0           RETURN FORM TRAP
01D0    2B48  DCLD 021     SKIP                             SET GA K=44              SET SRV-OUT
01D2    81CA  DCLD 022                      018  DECCNT      BR
01D4    7F3C  DCLD 023     RDFWD                            STB GB/IN V+1           STORE BUS IN,ADR +1
01D6    5006  DCLD 024                                      U=U-1                   DECREMENT COUNT BY 1
01D8    C4E3  DCLD 025                      030  CTZERO      BR IF Z=0               BR IF COUNT HAS GONE TO ZERO
01DA    0240  DCLD 026     CTNZ                             RTN G MMSKO=0           RETURN FORM TRAP
01DC    5F3C  DCLD 027     OUTPUT                           RDB GB/OUT V+1          RD OUT TO BUS-0,ADR+1
01DE    5006  DCLD 028                                      U=U-1                   COUNT -1
01E0    C4DA  DCLD 029                      026  CTNZ        BR IF ZNZ               BR IF COUNT IS NOT YET ZERO
01E2    C569  DCLD 030     CTZERO           033  CTCDA       BR IF G1 BIT0=1         BR IF CD FLAG
01E4    1573  DCLD 031                                      G1=G1*-K07              SET CT=0 INDICATION
01E6    0240  DCLD 032                                      RTN G MMSKO=0           RETURN FORM TRAP
01E8    3D00  DCLD 033     CTCDA                            SET GB K=80                 SET CDA REQ
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

01EA    0240     DCLD 034                                    RTN G MMSKO=C          RETURN FORM TRAP
01EC    2307     DCLD 035    EXPSTA                          V1=0
01EE    4F3F     DCLD 036                                    GB/OUT=V1              SET BUSS-OUT TO ZERO
01F0    2B42     DCLD 037                                    SET GA K=14            SET CMD OUT
01F2    D169     DCLD 038                  033   CTCDA       BR IF G1 BIT5=1        BR IF CMD 0 SET PREV
01F4    3543     DCLD 039                                    G1=G1$K04              G1 BIT5=1  CMD-0 GIVE
01F6    0240     DCLD 040                                    RTN G MMSKO=0          RETURN FORM TRAP
                                                        **********************************
                                                        * CROSS REFERENCE FOR CSECT DCLD *
                                                        **********************************

DCLD 011     DCLE 033
DCLD 013     DCLD 011
DCLD 018     DCLD 022  DCLR 140
DCLD 021     DCLD 016
DCLD 023     DCLD 015
DCLD 026     DCLD 029
DCLD 027     DCLD 014
DCLD 030     DCLD 019  DCLD 025
DCLD 033     DCLD 030  DCLD 038
DCLD 035     DCLD 013
```

DCLE DESCRIPTIVE TEXT

CHANNEL LOW PRIORITY TRAP STATUS ROUTINES

BYTE CHANNEL

WHEN THE BYTE CHANNEL OPTION IS LOADED IN CONTROL STORAGE, THIS ROUTINE HANDLES ALL TRAPS -- EITHER DATA SERVICE OR STATUS.

BYTE CHANNEL ENTRY POINTS

0180 -- LOW-PRIORITY TRAP-ADDRESS FOR DATA OR STATUS.

STATUS -- FROM ENTRY LOOP OF -DCLD- WHEN STATUS-IN RISES. THIS PORTION OF -DCLE- HANDLES STATUS.

BURST CHANNEL

WHEN THE BURST CHANNEL OPTION IS LOADED IN CONTROL STORAGE, THIS ROUTINE HANDLES TRAPS FOR EITHER --
1. DATA SERVICE - BUFFERED DEVICE, OR
2. STATUS - ANY DEVICE.

BURST CHANNEL ENTRY POINTS

0180 -- LOW-PRIORITY TRAP-ADDRESS AS DESCRIBED ABOVE.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              DCLE 001    T                    CHANNEL LOW PRIORITY TRAP STATUS ROUTINES   H.E.BERKEBILE
              DCLE 009    ATABLE  ADDR=0180
0180    2244  DCLE 010                                      LINK G MMSK2=1         TRAP WORD-SET PRIORTY
0182    C886  DCLE 011                          013  GO      BR IF GS BIT4=0        BR IF INL SELECT OFF
0184    0244  DCLE 012                                      RTN G MMSK2=0          RETURN FROM TRAP
0186    3613  DCLE 013    GO                                D0=D0$K01              SET SOURCE OF ENTRY INDICATOR
0188    CCA8  DCLE 014                          030  ADR40   BR IF GS BIT0=0        BR IF CDA REQ=0
018A    D127  DCLE 015                          029  ADR41   BR IF G1 BIT5=1        BR IF ICC DETECTED
018C    5212  DCLE 016                                      RDH V DA,8A            READ OUT NEXT CCW ADR
018E    5038  DCLE 017                                      RDH U V+2              READ CCW HWD 1
0190    008B  DCLE 018                                      Z=U0<K08               TEST COMMAND  FOR TIC
0192    F0A4  DCLE 019                          028  NOTIC   BR IF LZNZ             BR IF NOT A TIC
0194    5119  DCLE 020                                      U1=U1
0196    C4AC  DCLE 021                          026  PRGERD  BR IF ZNZ              BR IF TIC ADDR TOO BIG
0198    5230  DCLE 022                                      RDH V V+0              RD OUT NEW CCW ADDR
019A    0383  DCLE 023                                      Z=V1*-K08              CHECK FOR DOUBLE WORD BOUNDARY
019C    F0AC  DCLE 024                          026  PRGERD  BR IF LZNZ
019E    95F4  DCLE 025                   DCLB 081  RDHW1   BR
01A0    5212  DCLE 026    PRGERD                            RDH V DA,8A            READ OUT NEXT CCW ADR
01A2    95C2  DCLE 027                   DCLB 110  PRGERC  BR                     INCR CCW ADR IN DCLB &BR TO DCLR
01A4    95FA  DCLE 028    NOTIC   DCLB 084  NOTIC   BR                     BR TO DCLB & CONTINUE CCW FETCH
01A6    A154  DCLE 029    ADR41   DCLR 155  INFCHK  BR                     BR TO INTFCE CTRL CHECK RTNE
01A8    8BE6  DCLE 030    ADR40        032  ADR40B  BR
              DCLE 031    AEND
0BE6    EDE2  DCLE 032    ADR40B       034  ADR40A  BR IF GT BIT2=0       BR IF NO SRV-IN
0BE8    8172  DCLE 033                 DCLD 011  OPBRCH  BR
0BE2    FDEA  DCLE 034    ADR40A       036  NTSTIN  BR IF GT BIT3=0       BR IF STA-IN=0
0BE4    B09C  DCLE 035                 060  CHESTA  BR                     STATUS IS CHANNEL END TYPE
0BEA    2B40  DCLE 036    NTSTIN                    SET GA K=04                SET SEL OUT
0BEC    2645  DCLE 037                                      D0=0$K40               SET D0
0BEE    3613  DCLE 038                                      D0=D0$K01                 TO 41
0BF0    B0B6  DCLE 039                 040  ADR43   BR
30B6    CD87  DCLE 040    ADR43        048  ADR44   BR IF GT BIT0=1       BR IF ADR IN UP
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
30B8   262B    DCLE 041                                   D0=D0+K02          INCREMENT TMOUT COUNTER
30BA   CD87    DCLE 042                  048    ADR44      BR IF GT BIT0=1    BR IF ADR IN UP
30BC   E636    DCLE 043                  040    ADR43      BR IF D0 BIT2=0    BR IF NO TIMEOUT
30BE   DD9B    DCLE 044                  047    INFCHK     BR IF GT BIT1=1    BR IF NO SEL-IN
30C0   2800    DCLE 045                                   SET GA K=00        SEL-IN RECEIVED-DROP ALL TAGS OU
30C2   0244    DCLE 046                                   RTN G MMSK2=0           AND RETURN FROM TRAP
309A   A154    DCLE 047    INFCHK   DCLR 155  INFCHK     BR                 BR TO ERROR RTNES FOR ICC
3086   5F7F    DCLE 048    ADR44                          D1=GB/IN           D1=ADR ON BUSS-IN
3088   FC9B    DCLE 049                  047    INFCHK     BR IF GS BIT3=1    BR IF BAD PARITY ON BUSS-IN
308A   2007    DCLE 050                                   U0=0
308C   4F0F    DCLE 051                                   GB/OUT=U0          SET BUSS-OUT TO ZERO
308E   2802    DCLE 052                                   SET GA K=10             SET CMD-0---PROCEED
3090   E618    DCLE 053    ADR47         047    INFCHK     BR IF D0 BIT2=1    BR IF TIMEOUT
3092   FD99    DCLE 054                  057    ADR46      BR IF GT BIT3=1    BR IF STA-IN UP
3094   262B    DCLE 055                                   D0=D0+K02          INCREMENT  TIMEOUT  COUNTER
3096   FD90    DCLE 056                  053    ADR47      BR IF GT BIT3=0    BR IF STA-IN DWN
3098   A78E    DCLE 057    ADR46    DCLC 189  TRYIBS     BR                 TRY TO LOAD INT BUFFER
               DCLE 058  *                                                          **SET IB AND QUE
               DCLE 059  *                                                          **STATUS AT DEVICE
309C   FCA2    DCLE 060    CHESTA        063    ADR48      BR IF GS BIT3=0    BR IF DATA PARITY OK
309E   0D02    DCLE 061                                   RST GB K=10        RESET PARITY CHECK LATCH
30A0   3483    DCLE 062                                   G0=G0$K08          PUT CDC BITIN CHNLSTA
30A2   5439    DCLE 063    ADR48                          V1=G0              MOVE CHANNEL STATUS
30A4   C528    DCLE 064                  066    NCCRST     BR IF G1 BIT0=0    BR IF NO COA FLG
30A6   1565    DCLE 065                                   G1=G1*-K60         ZERO OUT CC AND  SLI FLGS
30A8   5F4F    DCLE 066    NCCRST                         G0=GB/IN           PUT GB/IN INTO G0
30AA   FC9B    DCLE 067                  047    INFCHK     BR IF GS BIT3=1    BR IF PARITY ERROR
30AC   E501    DCLE 068                  073    ADR50      BR IF G1 BIT2=1     BR IF SLI BIT=1
30AE   0583    DCLE 069                                   Z=G1*-K08           MASK G1-5,6,7
30B0   F081    DCLE 070                  073    ADR50      BR IF LZ=0          BR IF CLR
30B2   3345    DCLE 071                                   V1=V1$K40          PUTINCLENGTHINCHNLSTA
30B4   A782    DCLE 072    ADR51    DCLC 166  ADR11A     BR                 TRY TO SET INTRUPT BUFFER
3080   0313    DCLE 073    ADR50                          Z=V1*-K01          MASK CHANNEL STATUS
3082   C4B4    DCLE 074                  072    ADR51      BR IF ZNZ          BR IF CHNL STATUS NZ
3084   9ED2    DCLE 075               DCLC 140  ADRABC     BR                 CONTINUE TESTING STATUS
                                                ************************************
                                                * CROSS REFERENCE FOR CSECT DCLE *
                                                ************************************

DCLE 013    DCLE 011
DCLE 026    DCLE 021  DCLE 024
DCLE 028    DCLE 019
DCLE 029    DCLE 015
DCLE 030    DCLE 014
DCLE 032    DCLE 030
DCLE 034    DCLE 032
DCLE 036    DCLE 034
DCLE 040    DCLE 039  DCLE 043
DCLE 047    DCLE 044  DCLE 049  DCLE 053  DCLE 067
DCLE 048    DCLE 040  DCLE 042
DCLE 053    DCLE 056
DCLE 057    DCLE 054
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT DCLE *
                                  **********************************

DCLE 060     DCLE 035
DCLE 063     DCLE 060
DCLE 066     DCLE 064
DCLE 072     DCLE 074
DCLE 073     DCLE 068  DCLE 070
```

DCLH DESCRIPTIVE TEXT

HALT I/O ROUTINE

THIS ROUTINE EXECUTES A HALT I/O INSTRUCTION WHEN THE BURST CHANNEL OPTION IS LOADED IN CONTROL STORAGE.

ENTRY POINTS

ENTRY -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN A HALT I/O INSTRUCTION IS DECODED.

ENTRY POINTS CONTINUED

ADRO -- FROM -DCLT- FOR INACTIVE TEST I/O TO TRANSFER INFORMATION FROM ONE LOCAL STORAGE ZONE TO ANOTHER.

ADROA -- FROM -DCLT- FOR INACTIVE I/O INTERRUPT TO TRANSFER INFORMATION FROM ONE LOCAL STORAGE ZONE TO ANOTHER.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLH 001	T			CHANNEL	H.E.BERKEBILE
		DCLH 002	*			HALT I/O	
225E	C643	DCLH 004	ENTRY	016	ADR1	BR IF DO BIT0=1	BR IF ACTIVE
2260	2043	DCLH 005				U0=0$K04	SET UP U0T0 INDICATE HLT I/O
2262	7022	DCLH 006	ADRO			STH U DA,8C	TEMP STORE UNIT IDENT HWD
2264	24F6	DCLH 007	ADROA			SET MODE K=3F	SET CHNL MODE & ZONE
2266	2880	DCLH 008				SET GA K=08	SET INL SELECT LTCH
2268	5622	DCLH 009				RDH D DA,8C	RD OUT UNIT IDENT
226A	5659	DCLH 010				G1=DC	G1 = TST I/O OR HLT I/O IDENT
226C	2645	DCLH 011				D0=0$K40	SET D TO IDENTIFY BURST CHNL
226E	2407	DCLH 012				G0=0	G0 = CHNL STATUS
2270	D140	DCLH 013		015	HLTEBR	BR IF G15=0	BR IF NOT HALT I/O
2272	8DEA	DCLH 014		DCLC 020	HLTENT	BR	GO GET DEVICE OM INTERFACE
2240	8DF6	DCLH 015	HLTEBR	DCLC 018	NOPCIB	BR	GO GET DEVICE ON INTERFACE
2242	2C25	DCLH 016	ADR1			P0=0$K20	SET CC2
2244	24F6	DCLH 017				SET MODE K=3F	SET CHNL MODE
2246	2210	DCLH 018				SET MMSK K=01	SET DATA-TRAP PRIORITY
2248	F04E	DCLH 019		022	ADR6	BR IF G0 BIT7=0	BR IF NO STA RECD
224A	0210	DCLH 020				RST MMSK K=01	RST DATA-TRAP PRIORITY
224C	810C	DCLH 021		CCOM 155	CCOB	BR	BR TO CCO SET RTNE
224E	1D00	DCLH 022	ADR6			RST GB K=80	RST CDA REQ LTCH
2250	2B84	DCLH 023				SET GA K=28	SET ADR-OUT AND INL SEL LATCHES
2252	0210	DCLH 024				RST MMSK K=01	RST DATA-TRAP PRIORITY
2254	E55D	DCLH 025		029	NOINCL	BR IF G1 BIT2=1	BR IF SLI BIT ON
2256	4006	DCLH 026				U=U	U=COUNT
2258	C4DD	DCLH 027		029	NOINCL	BR IF Z=0	BR IF COUNT = C
225A	3445	DCLH 028				G0=G0$K40	PUT INC LGTH BIT IN CHNL STATS
225C	B02E	DCLH 029	NOINCL	DCLR 067	PRGECC	BR	BR TO RTNE TO SET INT BUFFER

****
* CROSS REFERENCE FOR CSECT DCLH *
****

DCLH 004 DCLA 123
DCLH 006 DCLT 005
DCLH 007 DCLT 064
DCLH 015 DCLH 013
DCLH 016 DCLH 004
DCLH 022 DCLH 019
DCLH 029 DCLH 025 DCLH 027

DCLL DESCRIPTIVE TEXT

ENTRY POINTS

IPL - ENTRY IS MADE TO THIS LABEL FROM THE -BSYS- ROUTINE. THE FOLLOWING STEPS HAVE TAKEN PLACE PRIOR TO ENTRY HERE -

1. THE LOAD KEY HAS BEEN PRESSED AND A MACHINE RESET PERFORMED.
2. THE RESIDENT DIAGNOSTIC -BDIA- WAS EXECUTED.
3. THE CHECK SUM ROUTINE -BCHK- HAS BEEN EXECUTED.
4. THE SYSTEM RESET ROUTINE -BSYS- HAS BEEN EXECUTED AND A BRANCH TO THIS ROUTINE AT LABEL * IPL * HAS BEEN PERFORMED.

IPLWTL - IF OPERATING AS (BURST CHANNEL), ENTRY IS MADE AT THIS LABEL FROM THE -DCLC- ROUTINE TO ENTER THE IPL WAIT LOOP AND WAIT FOR A TRAP TO ADDRESS 0170 IN THE -DCLD- ROUTINE, OR, IF BUFFERED DEVICE, TRAP TO ADDRESS 0180 IN THE -DCLE- ROUTINE.

IF OPERATING AS (BYTE CHANNEL), ENTRY IS MADE AT THIS LABEL FROM THE -DCLD- ROUTINE AFTER OPERATIONAL-IN HAS DROPPED. THE WAIT LOOP IS ENTERED TO WAIT FOR A TRAP TO ADDRESS 0180 IN THE -DCLE- ROUTINE.

IPLSTP - ENTRY IS MADE TO THIS LABEL BECAUSE OF SOME UNRECOVERABLE ERROR BEING DETECTED. A HARD STOP IS EXECUTED.

IPLSTB - ENTRY IS MADE TO THIS LABEL IF OPERATING (BURST CHANNEL), FROM THE -DCLC- ROUTINE TO CHECK CHANNEL STATUS AND THE COMMAND CHAINING FLAG.

IPLSTA - ENTRY IS MADE TO THIS LABEL IF OPERATING (BYTE CHANNEL), FROM THE -DCLC- ROUTINE AFTER ALL CCWS HAVE BEEN READ, TO STORE ZEROS IN THE FLAGS AND OP HALFWORD OF THE UCW.

PICKUP - ENTRY IS MADE TO THIS LABEL FROM EITHER THE 2540 REQUEST AND TEST I/O ROUTINE -ERRQ-, OR THE 2311 FEATURE, COMMAND VERIFICATION AND EXECUTION ROUTINE -FILX-. ENTRY IS MADE AFTER THE IPL IS COMPLETED FROM THE DEVICE TO STORE THE DEVICE ADDRESS IN PROGRAM STORAGE LOCATION 0002 AND BRANCH TO THE -BSYS- ROUTINE TO RESET IPL AND GO TO -BPSW- TO LOAD THE PSW.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLL 001	T			INITIAL PROGRAM LOAD (IPL)	H.E.BERKEBILE
3390	42A6	DCLL 002	IPL			V=T	T AND U ARE BOTH ZERO-FROM SYSR
3392	44A6	DCLL 003				G=T	
3394	2423	DCLL 004				G0=0$K02	G=0200 (READ CMD)
3396	74B8	DCLL 005				STH G T+2	MS--0000=0200
3398	70B8	DCLL 006				STH U T+2	MS--0002=0000
339A	2465	DCLL 007				G0=0$K60	G=6000 (CC&SLI FLGS)
339C	74B8	DCLL 008				STH G T+2	MS---0004=6000
339E	2115	DCLL 009				U1=0$K1C	
33A0	3183	DCLL 010				U1=U1$K08	U=0018 (COUNT)
33A2	70B8	DCLL 011				STH U T+2	MS---0006=0018
33A4	3B45	DCLL 012				T1=T1$K40	T=0048 FOR UNIT ADDRESS IN DCLA
33A6	46A6	DCLL 013				D=T	D=0048
33A8	17FD	DCLL 014				D1=D1<KF0	D=00B8
33AA	8ADA	DCLL 015		CSTS 071	RDSWCH	BAL	READ IPL ADDR
33AC	10F5	DCLL 016				U0=U0*-KF0	
33AE	24C3	DCLL 017				G0=0$K0C	
33B0	894E	DCLL 018		DCLA 095	UADRID	BR	
3488	0214	DCLL 019	IPLWTL			RST MMSK K=21	RESET MMSK 2
348A	848A	DCLL 024	WAITLP	024	WAITLP	BR	IPL WIAT LOOP

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT            COMMENTS

27D6    0000    DCLL 025   CIPLSTP                          XCTL'0000'
27D8    5007    DCLL 026    STOP                            STOP                  IPL STOP
27DA    A7D8    DCLL 027                        026  STOP    BR                   GO BACK TO STOP AGAIN
27DC    0313    DCLL 028    IPLSTB                          Z=V1*-K01
27DE    C4D6    DCLL 029                        025  IPLSTP  BR IF ZNZ            BR IF BAD CHNL STATUS
27E0    04C3    DCLL 031                                    Z=G0*-K0C
27E2    C4D6    DCLL 032                        025  IPLSTP  BR IF ZNZ            BR IF BAD STATUS
27E4    D557    DCLL 033                        025  IPLSTP  BR IF G1 BIT1=1      BR IF NO CC FLAG
27E6    0D04    DCLL 034                                    RST GB K=20              RST BURST LTCH
27E8    2407    DCLL 035                                    G0=0
27EA    7432    DCLL 036                                    STH G DA,8E          RESET  STATS  IN AUX0C8E
27EC    2B08    DCLL 037                                    SET GA K=40              SET SRV-0
27EE    C9EF    DCLL 038    OPINLP              038  OPINLP  BR IF GT BIT4=1      BR ON FALL OF OP-IN
27F0    0214    DCLL 039                                    RST MMSK K=21            RST MMSK 2
27F2    2400    DCLL 040    PICKUP                          SET MODE K=00
27F4    8ADA    DCLL 041    IPLADR  CSTS 071    RDSWCH      BAL                  READ IPL ADDR
27F6    10F5    DCLL 042                                    U0=U0*-KF0
27F8    2B23    DCLL 043                                    T1=0$K02
27FA    7080    DCLL 044                                    STH U T+0
27FC    A2E6    DCLL 045                BSYS 180    RETURN  BR                    GO RST IPL AND LOAD PSW
                                                            ***********************************
                                                            * CROSS REFERENCE FOR CSECT DCLL *
                                                            ***********************************

DCLL 002     BSYS 178
DCLL 019     DCLC 125
DCLL 024     DCLL 024
DCLL 025     DCHN 125  DCLC 060  DCLL 029  DCLL 032  DCLL 033  DCLR 003  DCLR 152
DCLL 026     DCLL 027  ERRQ 049  FILE 141  FILX 452
DCLL 028     DCLC 167
DCLL 038     DCLL 038
DCLL 040     ERRQ 048  FILX 458
```

DCLR DESCRIPTIVE TEXT

CHANNEL ERROR ROUTINES

ENTRY POINTS

PRGERR -- BURST CHANNEL ENTRY FROM -DCLB- WHENEVER A PROGRAM ERROR IS DETECTED WHILE FETCHING A CCW FOR A START I/O OR CHAINING.

PGESTT -- FROM -DCLA- WHEN A PROGRAM ERROR IS DETECTED WHILE FETCHING THE CAW FOR START I/O.

NTYPE -- FROM -BWRP- WHEN A STORAGE WRAP OR STORAGE PROTECT VIOLATION IS DETECTED WITH MODE SET FOR ONE OF THE INTEGRATED ATTACHMENTS.

PRGECC -- THIS BURST CHANNEL ENTRY LOADS THE INTERRUPT BUFFER, THEN THE MICROPROGRAM EITHER RETURNS FROM A TRAP OR GOES TO I-CYCLES.
ENTRY IS FROM ONE OF THE FOLLOWING.
1. -DCLR- IF PROGRAM ERROR IS DETECTED WHILE COMMAND CHAINING.
2. -DCLH- AFTER PERFORMING HALT I/O.

ENTRY POINTS CONTINUED

INTQD -- BYTE-CHANNEL ENTRY FROM -DCLC- WHEN THE INTERRUPT BUFFER IS FULL AND ANY CHANNEL ERROR OCCURS. THE DEVICE IS SELECTED AND THE STATUS IS QUEUED IN -DCLC-. IN -DCLR-, THE UCW IS UPDATED, THEN THE ROUTINE BRANCHES TO -DCLD- TO EITHER RETURN TO I-CYCLES OR TO RETURN FROM TRAP.

WRAP
PROTCH -- FROM -BWRP- TO SET EITHER PROGRAM CHECK OR PROTECTION CHECK IN STATUS, AND SET UP TO STOP THE DEVICE ON THE NEXT SERVICE-IN.

INFCHK -- COMMON ENTRY FOR INTERFACE CONTROL CHECK LOG-OUT. THE INTERFACE IS CLEARED AND THE CHANNEL ERROR STATUS IS SET UP. THEN THE ROUTINE EITHER RETURNS FROM THE TRAP OR GOES TO I-CYCLES, DEPENDING ON SOURCE OF ENTRY.

CCCXXX -- FROM -BMCK- AFTER MACHINE CHECK TRAP IF MMSK BITS INDICATE A CHANNEL CONTROL CHECK.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT                COMMENTS
                DCLR 001    T                              CHANNEL ERROR ROUTINES                  '    H.E. BERKEBILE
3002    C3C0    DCLR 002     PRGERR          004  NOIPLM     BR IF BA4=0              BR IF NOT IPL
3004    A7D6    DCLR 003                  DCLL 025  IPLSTP    BR                       GO TO IPL STOP
3040    F207    DCLR 004     NOIPLM           013  PGECHN     BR IF D07=1              BR IF FROM CHAINING
3042    2400    DCLR 008     PGESTT                          SET MODE K=00
3044    2F25    DCLR 009                                     H1=0$K20                 SET PROGRAM CHECK IN CHNL STATUS
3046    2E05    DCLR 010     ADR003                          H0=0$K00                 H0=00 UNIT STA  H1=CHNL STA
3048    A030    DCLR 011                  CCOM 135  STATUS    BR                       STORE STATUS ONLY CSW-SET CC1
3006    D623    DCLR 013     PGECHN           061  BURST      BR IF D0 BIT1=1          BR IF BURSTCH
3008    1773    DCLR 019                                     D1=D1*-K07               SET D TO UCW HWD 0
300A    8F7A    DCLR 030     NOTHEM      ERRQ 082  RDCHST     BAL                      GET CHAN AND UNIT STATUS IN G,H
300C    C510    DCLR 031                     033  NODATC     BR IF G10=0              BR IF NO DATA CHAIN
300E    2EC3    DCLR 032                                     H0=0$K0C                 SET CHANNEL END AND DEVEND
3010    15C5    DCLR 033     NODATC                          G1=G1*-KC0               RESET CHAIN FLAGS
3012    34A5    DCLR 034                                     G0=G0$KA0                SET SEC AND PROG CHECK
3014    C398    DCLR 036                     043  NATPRO     BR IF BA4=0              BR IF NOT IPL
3016    9F5C    DCLR 038                ERRQ 049  STOP       BR                       GO TO 2540 STOP
3018    E71E    DCLR 043     NATPRO          052  NTYPE      BR IF D12=0              CHECK FOR
301A    F71E    DCLR 047                     052  NTYPE      BR IF D13=0               1052
301C    B478    DCLR 048                DYPE 237  PRGCHK     BR
301E    B43C    DCLR 052     NTYPE      ERRQ 109  GUESS      BAL                      RST 2540 CMD.INTLK.
3020    8D68    DCLR 053                DPTR 075  TAKEIN     BR        **TRY 1403,2540 INTRPT--GO TO SOFT STOP
3022    2425    DCLR 061     BURST                           G0=0$K20                 SET PROGRAM CHECK
3024    54E9    DCLR 062                                     H0=G0                    MOVE CHANNEL STATUS
3026    15C5    DCLR 063     ADRXYZ                          G1=G1*-KC0               RST CDA & CC FLAGS
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
3028   1573   DCLR 064                                    G1=G1*-K07         RST  CODED OP BITS
302A   CCAE   DCLR 065                 067     PRGECC     BR IF GS BIT0=0    BR IF CDA FLAG OFF
302C   9660   DCLR 066                DCLB 223   BSTCDA     BR                 BR TO DCLB TO RTN FROM TRAP
302E   0D10   DCLR 067   PRGECC                            RST GB K=01            RESET SUPR-0
3030   3413   DCLR 068                                    G0=G0$K01          SET STATUS RECEIVED INDICATOR
3032   2205   DCLR 069                                    V0=0$K00           SET CC UNIT STATUS
3034   7602   DCLR 070                                    STH D DA,88        STORE UNIT IDENT HWD IN IB
3036   2D40   DCLR 071                                    SET GB K=04        SET INT BUFFER LATCH
3038   C681   DCLR 072                 076     HIPRTP     BR IF MMSK0=1      BR IF CHNL HI PRIORITY TRAP
303A   E681   DCLR 073                 076     HIPRTP     BR IF MMSK2=1      BR IF CHNL LO TRP PRIORITY
303C   2800   DCLR 074                                    SET GA K=00        RST TAGS OUT
303E   A7A4   DCLR 075                DCLC 201   ADR22A     BR                 BR TO SET CMD-0 &TMOUT ON OPI DW
3000   0240   DCLR 076   HIPRTP                           RTN G MMSK0=0      RTN FROM TRAP
3490   3425   DCLR 134   WRAP                             G0=G0$K20          SET PROGRAM CHECK-MEMORY WRAP
3492   98D8   DCLR 135                 137     WRPPRT     BR
18D6   3415   DCLR 136   PROTCH                           G0=G0$K10          SET PROTECTION CHECK
18D8   15C5   DCLR 137   WRPPRT                           G1=G1*-KC0         RST CDA & CC FLAGS
18DA   1533   DCLR 138                                    G1=G1*-K03         RST CODED COMMAND BITS
18DC   81CA   DCLR 140                DCLD 018   DECCNT     BR
218E   A7D6   DCLR 152   IPLICC     DCLL 025   IPLSTP     BR                 BR TO IPL STOP WORD
2154   562F   DCLR 155   INFCHK                           V0=MMSK            SAVE MMSK REG
2156   221E   DCLR 156                                    SET MMSK K=71          SET PRIORITY HOLD LATCH
2158   5C3F   DCLR 157                                    V1=GS              SAVE GS REG
215A   7222   DCLR 159                                    STH V DA,8C        TEMP STORE MMSK/GS
215C   5D2F   DCLR 163   ADR102                          V0=GT              SAVE GT EXTERNAL CONS
215E   5E3F   DCLR 164                                    V1=GD              SAVE GD EXTERNAL CONS
2160   2B10   DCLR 165                                    SET GA K=01        SET STP TRAP INHIBIT LATCH
2162   2407   DCLR 166                                    G0=0
2164   2585   DCLR 167                                    G1=0$K80
2166   3573   DCLR 168                                    G1=G1$K07          G=0087
2168   765A   DCLR 169                                    STH D G-2          STORE UNIT IDENT HWD IN MS 0086
216A   725A   DCLR 170                                    STH V G-2          STORE GT/GD IN MS0084
216C   5350   DCLR 171                                    RDB V1 G+0             READ OUT ERROR COUNT INTO V1
216E   231B   DCLR 172                                    V1=V1+K01          INCREMENT ERROR COUNT
2170   33C5   DCLR 173                                    V1=V1$KC0          SET  'LGOUT WAS CHNL ICC' BIT
2172   5622   DCLR 175                                    RDH D DA,8C        READ OUT MMSK/GS
2174   5729   DCLR 179                                    V0=D1              MOVE GS TO V0 REG
2176   725A   DCLR 180                                    STH V G-2          STORE GS/ERROR COUNT IN MS0082
2178   577F   DCLR 181                                    D1=BA              MOVE BA TO D1 REG
217A   7650   DCLR 182                                    STH D G            STORE MMSK/BA IN MS0080
217C   3573   DCLR 183                                    G1=G1$K07          BUILD 0087 ADDRESS
217E   5650   DCLR 184                                    RDH D G            RESTORE UNIT IDENT HWD IN D
2180   E690   DCLR 188                 191     ADR100     BR IF MMSK2=0      BR IF NOT IN TRAP
2182   2B00   DCLR 189                                    SET GA K=00        RST CHNL TAGS
2184   A194   DCLR 190                 193     ADR101     BR
2190   24F6   DCLR 191   ADR100                          SET MODE K=3F
2192   2B80   DCLR 192                                    SET GA K=08        SET INL SELECT LATCH
2194   2507   DCLR 193   ADR101                          G1=0               ZERO OUT G1
2196   021E   DCLR 194                                    RST MMSK K=71
2198   2323   DCLR 195                                    V1=0$K02           SET INTERFACE CONTROL CHECK
219A   0D02   DCLR 196                                    RST GB K=10        RESET PARITY ERROR LATCH
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
219C    C38F     DCLR 197   IPLICK        152  IPLICC     BR IF BA BIT4=1    BR IF IPL
219E    2B80     DCLR 198                                 SET GA K=08        SET INL SELECT LATCH
21A0    C9B8     DCLR 199                   212  OPIDWN     BR IF GT BIT4=0    BR IF OP-IN IS ALREADY DOWN
21A2    2B84     DCLR 200                                 SET GA K=28        SET ADR-OUT AND INL SEL LTCHES
21A4    2207     DCLR 201                                 V0=0               INITIALIZE TIMEOUT COUNTER
21A6    C9B8     DCLR 202   ADR001             212  OPIDWN     BR IF GT BIT4=0    BR IF OP-IN DOWN
21A8    221B     DCLR 203                                 V0=V0+K01          INCREMENT TMOUT COUNTER
21AA    F4A6     DCLR 204                   202  ADR001     BR IF AC=0         BR IF NO TIMEOUT
21AC    2D10     DCLR 205                                 SET GB K=01            SET SUPR-OUT
21AE    3B80     DCLR 206                                 SET GA K=88        DROP OP-OUT
21B0    221B     DCLR 207   WTOPI2                        V0=V0+K01          INCREMENT TIMEOUT COUNTER
21B2    C9B8     DCLR 208                   212  OPIDWN     BR IF GT BIT4=0    BR IF OP-IN DOWN
21B4    F4B0     DCLR 209                   207  WTOPI2     BR IF AC=0         BR IF NO TIMEOUT
21B6    CD10     DCLR 210                                 RST GB K=01            RST SUPR-OUT
21B8    C687     DCLR 212   OPIDWN             229  CHAING     BR IF MMSK0=1      BR IF CHNL HI TRP PRIORITY
21BA    E687     DCLR 213                   229  CHAING     BR IF MMSK2=1      BR IF CHNL LO TRP PRIORITY
21BC    F207     DCLR 217                   229  CHAING     BR IF D0 BIT7=1    BR IF ENTRY WAS VIA CHAINING
21BE    221E     DCLR 218                                 SET MMSK K=71      SET PRIORITY HOLD
21C0    7222     DCLR 219                                 STH V DA,8C        TEMP STORE UNIT & CHNL STATUS
21C2    D8C6     DCLR 220                   222  INTOFF     BR IF GS BIT5=0    BR IF INTERRUPT LATCH OFF
21C4    3413     DCLR 221                                 G0=G0$K01          SET STATUS RECEIVED BIT
21C6    2B00     DCLR 222   INTOFF                        SET GA K=00        DROP ALL TAGS OUT
21C8    2400     DCLR 223                                 SET MODE K=00      SET CPU MODE & ZONE
21CA    5E22     DCLR 224                                 RDH H DA,8C        READ OUT UNIT/CHNL STATUS
21CC    021F     DCLR 225                                 RST MMSK K=71      RESET PRIORITY HOLD
21CE    1000     DCLR 226                                 RST S0             RST RETURN FLAG IF ON
21D0    B046     DCLR 227                   010  ADR003     BR                 BR TO STORE CSW (STATUS ONLY)
2186    5349     DCLR 229   CHAING                        G0=V1              MOVE CHNL STATUS
2188    3613     DCLR 230                                 D0=D0$K01          SET "SOURCE OF ENTRY WAS CHAIN"
218A    1D00     DCLR 231                                 RST GB K=80        RESET CDA REQ LTCH
218C    B026     DCLR 232                   063  ADRXYZ     BR
18F8    2343     DCLR 238   CCCXXX                        V1=0$KC4           SET CHNL CONTROL CHECK
18FA    5612     DCLR 239                                 RDH D DA,8A        RD OUT UNIT IDENT HWD
18FC    3613     DCLR 240                                 D0=D0$K01          SET "SOURCE OF ENTRY WAS CHAIN"
18FE    A19C     DCLR 241                   197  IPLICK     BR
                                                  **********************************
                                                  * CROSS REFERENCE FOR CSECT DCLR *
                                                  **********************************

DCLR 002    DCLB 105  DCLB 108
DCLR 004    DCLR 002
DCLR 008    DCLA 074
DCLR 010    DCLR 227
DCLR 013    DCLR 004
DCLR 033    DCLR 031
DCLR 043    DCLR 036
DCLR 052    BWRP 156  BWRP 157  DCLR 043  DCLR 047
DCLR 061    DCLR 013
DCLR 063    DCLR 232
DCLR 067    DCLH 029  DCLR 065
DCLR 076    DCLR 072  DCLR 073
DCLR 134    BWRP 046
```

****
* CROSS REFERENCE FOR CSECT DCLR *
****

```
DCLR 136     BWRP 079
DCLR 137     DCLR 135
DCLR 152     DCLR 197
DCLR 155     DCLC 038  DCLC 124  DCLC 191  DCLE 029  DCLE 047
DCLR 191     DCLR 188
DCLR 193     DCLR 190
DCLR 197     DCLR 241
DCLR 202     DCLR 204
DCLR 207     DCLR 209
DCLR 212     DCLR 199  DCLR 202  DCLR 208
DCLR 222     DCLR 220
DCLR 229     DCLR 212  DCLR 213  DCLR 217
DCLR 238     BMCK 065
```

DCLT DESCRIPTIVE TEXT

CHANNEL TEST I/O AND I/O INTERRUPT

BURST CHANNEL ENTRY POINTS

TESTIO -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN A TEST I/O INSTRUCTION IS DECODED.

INTRTX -- FROM -DCLC- TO RESTORE THE PREVIOUS CONTENTS OF THE H-REGISTER BEFORE THE MICROPROGRAM RETURNS TO I-CYCLES.

INTENT -- FROM -BSWI- WHEN AN I/O INTERRUPT IS DETECTED.

BYTE CHANNEL ENTRY POINTS

ENTRY1 -- FROM -DCLA- I/O INSTRUCTIONS ROUTINE WHEN AN ACTIVE TEST I/O IS DECODED.

ENTRY2 -- FROM -DCLA- WHEN AN INACTIVE TEST I/O IS DECODED.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		DCLT 001	T			CHANNEL TEST I/O AND I/O INTERRUPT	H.E.BERKEBILE
1C28	C62F	DCLT 003	TESTIO	009	ACTIVE	BR IF DO BIT0=1	BR IF ACTIVE BIT ON
1C2A	2007	DCLT 004				U0=0	SET U0 FOR TEST I/O
1C2C	A262	DCLT 005		DCLH 006	ADR0	BR	GO SET UP MODE AND ZONE
		DCLT 006	*				RETREIVE STATUS FROM
		DCLT 007	*				UNIT
1C26	AFCA	DCLT 008	ADR0	CCOM 162	CC2B	BR	BR TO CC2 SET
1C2E	24F6	DCLT 009	ACTIVE			SET MODE K=3F	SET CHNL MODE & ZONE
1C30	ECA6	DCLT 010		008	ADR0	BR IF GS2=0	** BR IF FILE IS ACTIVE-NOT CHANNEL
1C32	F026	DCLT 011		008	ADR0	BR IF G0 BIT7=0	BR IF NO STATUS RECEIVED
1C34	2400	DCLT 012				SET MODE K=00	SWITCH BACK TO CPU MODE & ZONE
1C36	5202	DCLT 013				RDH V DA,88	READ INT BUFFER
1C38	6311	DCLT 014				V1=V1<U1	ADDRESS MATCH
1C3A	C4A6	DCLT 015		008	ADR0	BR IF ZNZ	BR IF NO MATCH
1C3C	24F6	DCLT 016				SET MODE K=3F	SET BACK TO CHNL MODE & ZONE
1C3E	C142	DCLT 017	ADR1	019	SKIP	BR IF G14=0	BR IF NO PCI
1C40	3485	DCLT 018				G0=G0$K80	SET PCI
1C42	70F2	DCLT 019	SKIP			STH U DA,BE	SAVE COUNT
1C44	5419	DCLT 020				U1=G0	MOVE STATUS
1C46	5209	DCLT 021				U0=V0	UNIT STATUS
1C48	4452	DCLT 022				RDH G DC,9A	READ ZERO CONSTANT INTO G
1C4A	7432	DCLT 023				STH G DA,8E	ZERO OUT UNIT ADDRESS
1C4C	0D04	DCLT 024				RST GB K=20	RST BURST LATCH
1C4E	70E2	DCLT 025	REDUC			STH U DA,BC	STORE STATUS
1C50	0D40	DCLT 026				RST GB K=04	RST INT
1C52	021E	DCLT 027	SAVE1			RST MMSK K=71	ALLOW TRAPS
1C54	2400	DCLT 028				SET MODE K=00	
1C56	4652	DCLT 029				RDH D DC,9A	READ ZERO CONSTANT INTO D
1C58	2B87	DCLT 031				T1=0$K88	GET
1C5A	52AC	DCLT 032				ISK STP0 AS,T+2	PROTECT
1C5C	526F	DCLT 033				D0=STP0	KEY
1C5E	566B	DCLT 034				D0=D0H	
1C60	5212	DCLT 036				RDH V DA,8A	CCW ADDRESS
1C62	50F2	DCLT 037				RDH U DA,BE	COUNT
1C64	5EE2	DCLT 038				RDH H DA,BC	STATUS
1C66	1F13	DCLT 039				H1=H1*-K01	RST INT. BIT

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS
1C68    A024    DCLT 040              CCOM 127  CSWADR     BAL                   STORE CSW SO=1FOR INT.RTN
1C6A    8C7A    DCLT 041    INTRTX    CCOM 187  RESTRH     BAL                  RESTORE CPU H1
1C6C    5A22    DCLT 042                                   RDH T DA,8C          UNIT ADDRESS FOR INT.
1C6E    2A13    DCLT 043                                   TO=0$K01              FIX CHANNEL ADDRESS
1C70    2400    DCLT 044    INTRTN                         SET MODE K=00
1C72    A43E    DCLT 045              DCHN 105  USECOM     BR                   GO TAKE INTERRUPT
                DCLT 046  *                                 ****************************************
2218    3000    DCLT 047    INTENT                         SET SO               SET THE RETURN FLAG
221A    2480    DCLT 049                                   SET MODE K=08                 FILE MODE
221C    5432    DCLT 050                                   RDH G DA,8E
221E    C425    DCLT 051                   055  ACTON      BR IF GO BIT0=1
2220    E9A8    DCLT 052                   060  INACHL     BR IF FFI6=0
2222    A6FC    DCLT 053    FINT      FINT 002  INTRPT     BR
                DCLT 054  *                                 ****************************************
2224    D9A3    DCLT 055    ACTON          053  FINT       BR IF FFI5=1
2226    C9A3    DCLT 056                   053  FINT       BR IF FFI4=1         BR IF PCI LATCH IS ON
2228    5A02    DCLT 060    INACHL                         RDH T DA,88          RD UNIT IDENT HWD FROM IB
222A    2A15    DCLT 061                                   TO=0$K10             SET TO TO TEST I/O INDICATOR
222C    7A22    DCLT 062                                   STH T DA,8C          TEMP STORE UNIT ADDRESS
222E    2A13    DCLT 063                                   TO=0$K01             SET TO TO CHNL ADDRESS
2230    C464    DCLT 064              DCLH 007  ADROA      BR IF GO BIT0=0      BR IF NOT ACTIVE
2232    2A07    DCLT 065                                   TO=0
2234    2413    DCLT 066                                   GO=0$K01             SET GO FOR INTERRUPT
2236    24F6    DCLT 067                                   SET MODE K=3F        SET CHNL MODE & ZONE
2238    221F    DCLT 068                                   SET MMSK K=71        SET PRIORITY HOLD
223A    F074    DCLT 069                   072  ACTPCI     BR IF GO BIT7=0       BR IF NOT FROM INT
223C    1413    DCLT 070                                   GO=GO*-K01           RESET GO BIT7
223E    9C3E    DCLT 071                   017  ADR1       BR                   BR TO CSW STORE
2274    1583    DCLT 072    ACTPCI                         G1=G1*-K08           TURN OFF PCI IN OPFLG
2276    FCFA    DCLT 073                   075  NOCDZ      BR IF GS3=0           BR IF NO PARITY ERROR
2278    3483    DCLT 074                                   GO=GO$K08             SET DATA CHECK
227A    0D40    DCLT 075    NOCDZ                          RST GB K=04          RST INT. LATCH
227C    2470    DCLT 076                                   SET MODE K=07         SET CHANNEL ZONE-CPU MODE
227E    74A2    DCLT 077                                   STH G DA,AC          STORE STATUS
2280    70F2    DCLT 078                                   STH U DA,BE          STORE COUNT
2282    2400    DCLT 079                                   SET MODE K=00
2284    5BA2    DCLT 080                                   RDB T1 DA,AC         GET CHAN STATUS
2286    3B85    DCLT 081                                   T1=T1$K80            SET PCI
2288    7AE2    DCLT 082                                   STH T DA,BC          STORE STATUS WITH PCI
228A    9C52    DCLT 083                   027  SAVE1      BR
                                           **********************************
                                           * CROSS REFERENCE FOR CSECT DCLT *
                                           **********************************

DCLT 003     DCLA 122
DCLT 008     DCLT 010  DCLT 011  DCLT 015
DCLT 009     DCLT 003
DCLT 017     DCLT 071
DCLT 019     DCLT 017
DCLT 027     DCLT 083
DCLT 041     DCLC 058
DCLT 047     BSWI 134
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT DCLT *
                                  **********************************

DCLT 053     DCLT 055  DCLT 056
DCLT 055     DCLT 051
DCLT 060     DCLT 052
DCLT 072     DCLT 069
DCLT 075     DCLT 073
```

DCOM -- COMMON I/O ROUTINES

THESE COMMON ROUTINES ARE EMPLOYED BY THE I/O ROUTINES FOR THE NATIVE DEVICES -- PRINTER, READER-PUNCH, AND PR-KB.

ENTRIES

PCIBR -- USED TO TRY FOR A PROGRAM-CONTROLLED INTERRUPT. IF THE PCI FLAG IS OFF, THE MICROPROGRAM RETURNS TO THE LINK ADDRESS. IF THE PCI FLAG IS ON, THE PCI-ALONE INDICATOR BIT IS SET FOR THE INTERRUPT BUFFER. FROM THIS POINT, PCI IS HANDLED THE SAME AS THE NORMAL INTERRUPT.

INTRPT -- NORMAL INTERRUPT ENTRY, I.E., WHEN CHANNEL-END AND/OR DEVICE-END STATUS IS READY. IF THE INTERRUPT IN I.B. BIT IS ON, THE DEVICE ALREADY HAS CONTROL OF THE I.B. OTHERWISE, THE ROUTINE MUST TEST TO SEE IF THE I.B. IS AVAILABLE. IF THE I.B. IS NOT AVAILABLE, THE INTERRUPT IS STACKED AND ANOTHER REQUEST WILL OCCUR WHEN THE I.B. IS AVAILABLE.
IF THE I.B. IS AVAILABLE, AN INTERRUPT IS INITIATED AS FOLLOWS.
1. SET CHANNEL 0 INTERRUPT LATCH, S7.
2. SET I.B. BIT IN CHANNEL-STATUS REG.
3. STORE UCW ADDRESS IN INTERRUPT BUFFER.
4. STORE PCI-ALONE BIT IN I.B. IF THIS BIT WAS TURNED ON AT -PCIBR- ENTRY.

AFTER THE INTERRUPT IS INITIATED, THE MICRO-PROGRAM RETURNS TO THE LINK ADDRESS.

CHANOK
REDSTR -- USED TO UPDATE THE UCW. CHANNEL STATUS, FLAGS/OP, UNIT STATUS, AND COUNT ARE STORED IN THE UCW. IF COMMAND CHAINING, AND DEVICE-END IS SET, ENTRY IS TO -CHANOK- TO CLEAR THE UNIT STATUS REGISTER.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS

                DCOM 001    T                  COMMON I/O ROUTINES
                DCOM 002    *
                DCOM 003    *                       THIS ROUTINE TRIES TO PUT THE NATIVE DEVICE INTO THE INTERRUPT
                DCOM 004    *                       BUFFER FOR PCI OR STATUS.
                DCOM 005    *
OFBA    C174    DCOM 006     PCIBR          061  RETURN    BR IF G14=0            BR IF NO PCI
OFBC    2683    DCOM 007                                   D0=0$K08               SET PCI ALONE
OFBE    F047    DCOM 008     INTRPT         012  ITSME     BR IF G07=1              BR IF SAME IN INT BUFFER
OFC0    221E    DCOM 009                                   SET MMSK K=71            BLOCK TRAPS
OFC2    C7E9    DCOM 010                    045  IBTAKN    BR IF BA BIT0=1        BR IF IB FULL
OFC4    2010    DCOM 011                                   SET S7                 TAKE IT
OFC6    76D2    DCOM 012     ITSME                         STH D DA,BA            STORE ADDRESS IN IB
OFC8    021E    DCOM 013                                   RST MMSK K=71          ALLOW TRAPS
OFCA    3413    DCOM 014                                   G0=G0$K01              SET IB BIT
OFCC    1683    DCOM 015                                   D0=DC*-K08
OFCE    F75B    DCOM 025     RSTREQ         030  RDTW1     BR IF D1 BIT3=1        RST REQUEST, BR IF RDR OR TW
OFD0    E757    DCOM 027                    034  PCH1      BR IF D1 BIT2=1        BR IF PCH
OFD2    CD08    DCOM 028                                   RST PRA K=40             RST PTR REQUEST
OFD4    128E    DCOM 029                                   RTN
OFDA    E761    DCOM 030     RDTW1          041  TW1       BR IF D1 BIT2=1        BR IF TW
OFDC    0B20    DCOM 032                                   RST R K=02              RST RDR REQUEST
OFDE    128E    DCOM 033                                   RTN
OFD6    0F20    DCOM 034     PCH1                          RST P K=02             RST PCH REQUEST
OFD8    128E    DCOM 040                                   RTN
OFE0    0F24    DCOM 041     TW1                           RST TA K=22             RST SHARE MICRO FORCE
OFE2    C866    DCOM 042                    044  TYREXT    BR IF P04=0
OFE4    1C83    DCOM 043                                   P0=P0*-K08             RST END STAT
OFE6    128E    DCOM 044     TYREXT                        RTN
OFE8    021E    DCOM 045     IBTAKN                        RST MMSK K=71           SOMEONE ELSE IN IB RST MMSK
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
OFEA    1683    DCOM 046                                   D0=D0*-K08           RST PCI ALONE
OFEC    D575    DCOM 047                 061     RETURN    BR IF G11=1           DONT QUE PCI CMND CHAIN
OFEE    F77B    DCOM 057                 062     RDTW2     BR IF D1 BIT3=1     SET QUE AND REQ.,&R IF RDR OR TW
OFF0    F777    DCOM 059                 066     PCH2      BR IF D1 BIT2=1     BR IF PCH
OFF2    2D88    DCOM 060                                   SET PRA K=48        SET PTR REQUEST AND QUE
OFF4    128E    DCOM 061    RETURN                         RTN
OFFA    E735    DCOM 062    RDTW2            073     TW2       BR IF D1 BIT2=1     BR IF TW
OFFC    2B30    DCOM 064                                   SET R K=03          SET RDR REQUEST AND QUE
OFFE    128E    DCOM 065                                   RTN
OFF6    2F30    DCOM 066    PCH2                           SET P K=03          SET PCH REQUEST AND QUE
OFF8    128E    DCOM 072                                   RTN
OFB4    C838    DCOM 073    TW2              075     TWREXT    BR IF P04=0
OFB6    2F04    DCOM 074                                   SET TA K=20          SET 1052 MICRO FORCE
OFB8    128E    DCOM 075    TWREXT                         RTN
                DCOM 076   *
                DCOM 077   *                    THIS ROUTINE STORES THE CHNL STATUS, FLAGS, UNIT STATUS AND
                DCOM 078   *                    THE COUNT OF THE NATIVE DEVICE INTO ITS UCW.
                DCOM 079   *
3404    2EC7    DCOM 080    CHANOK                         H0=0
3406    7468    DCOM 081    REDSTR                         STH G AS,D+2
3408    7E68    DCOM 082                                   STH H AS,D+2
340A    7060    DCOM 083                                   STH U AS,D+0        STORE COUNT
340C    C511    DCOM 084                 086     RETUN     BR IF G1 BIT0=1     BR IF CDA
340E    2A07    DCOM 085                                   T0=0                RST T0
3410    128E    DCOM 086    RETUN                          RTN
```

```
                                ********************************
                                * CROSS REFERENCE FOR CSECT DCOM *
                                ********************************
DCOM 006    DPTR 069  DYPE 040
DCOM 008    DPTR 075  DYPE 013  DYPE 086
DCOM 012    DCOM 008
DCOM 025    ERRQ 037  ERRQ 051
DCOM 030    DCOM 025
DCOM 034    DCOM 027
DCOM 041    DCOM 030
DCOM 044    DCOM 042
DCOM 045    DCOM 010
DCOM 061    DCOM 006  DCOM 047
DCOM 062    DCOM 057
DCOM 066    DCOM 059
DCOM 073    DCOM 062
DCOM 075    DCOM 073
DCOM 080    DPTR 073  DPTS 142
DCOM 081    DPTR 076  DYPE 238  ERRQ 047
DCOM 086    DCOM 084
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS

              DMCS 001     T           MCS LOAD ROUTINE  KREPPEL
              DMCS 002     *
237E   1E7D   DMCS 003      CD737B                        H0=H0<K70
2380   E09C   DMCS 004                    022  CMDREJ     BR IF HZNZ         COMMAND REJECT IF NOT 7X CMD
2382   E81C   DMCS 005                    022  CMDREJ     BR IF P06=0        COMMAND REJECT IF NOT 73 OR 7B
2384   EB09   DMCS 006                    009  RSTBIT     BR IF H16=1        BR IF GATE LOAD TAG IS ON
2386   F21D   DMCS 007                    022  CMDREJ     BR IF D07=1        BR IF NOT FIRST CMD IN CHAIN
2388   2C75   DMCS 009      RSTBIT                        P0=0$K70           SET XLATE TABLE ADDR FOR NO 24K
238A   2D05   DMCS 013                                    P1=0$K00           SET ADDR TO 7X00 OR 9X00
238C   53C0   DMCS 014                                    RDB V1 AS,P        READ PCCL CONSTANT
238E   1385   DMCS 015                                    V1=V1*-K80         RESET BLOCK DATA CHECK BIT
2390   CA15   DMCS 016                    018  ADACHK     BR IF H04=1        BR IF ALLOW DATA CHK *7B CMD*
2392   3385   DMCS 017                                    V1=V1$K80          SET BLK DATA CHK BIT *73 CMD*
2394   73C0   DMCS 018      ADACHK                        STB V1 AS,P        RESTORE PCCL CONSTANT
2396   8C58   DMCS 019                DPTS 125  NOOP      BR                 GO TO NO-OP
2398   EE4A   DMCS 020      EBF3FB        023  GATELD     BR IF H02=0        BRANCH IF EB **GATE LOAD**
239A   EB1F   DMCS 021                    026  LOADOK     BR IF H16=1        BR IF GATE LOAD TAG **F3 OR FB**
239C   8C5C   DMCS 022      CMDREJ   DPTS 131  CMDREJ     BR                 GO TO COMMAND REJECT
23CA   F21D   DMCS 023      GATELD        022  CMDREJ     BR IF D07=1        BR IF NOT FIRST CMD IN CHAIN
23CC   3F23   DMCS 024                                    H1=H1$K02          SET GATE LOAD TAG
23CE   8C58   DMCS 025                DPTS 125  NOOP      BR                 GO TO NO-OP
239E   74E2   DMCS 026      LOADOK                        STH G DA,BC        SAVE G REG
23A0   76F2   DMCS 027                                    STH D DA,BE        SAVE D REG
23A2   2F10   DMCS 028                                    SET PRB K=01       SET DIAGNOSTIC DECODE 4
              DMCS 029     *
              DMCS 030     *           DETERMINE  CHAIN  LENGTH  ROUTINE
              DMCS 031     *
23A4   2713   DMCS 032                                    D1=0$K01                BUILD INITIAL PCCL CONSTANT
23A6   72A2   DMCS 033                                    STH V DA,AC             SAVE INITIAL DATA ADDRESS
23A8   2427   DMCS 034                                    G0=0$K22                BUILD ADDRESS OF THE
23AA   25A3   DMCS 035                                    G1=0$K0A                  TABLE OF CHAIN LENGTHS
23AC   4148   DMCS 036      TRYLEN                        RDB U1 CS,G+1           READ COUNTER LENGTH FROM TABL
23AE   51C9   DMCS 037                                    P0=U1                   SET COUNTER TO CHAIN LENGTH
23B0   4826   DMCS 038                                    I=V                     BUILD ADDRESS
23B2   691B   DMCS 039                                    I1C=I1+U1                 EQUAL TO DATA ADDRESS
23B4   68AD   DMCS 040                                    I0C=I0+T0+C                 PLUS THE CHAIN LENGTH *T0=0*
23B6   5B38   DMCS 041      COMPAR                        RDB T1 V+1              READ DATA BYTE 1
23B8   5D98   DMCS 042                                    RDB P1 I+1              READ DATA BYTE 2
23BA   2C1B   DMCS 043                                    P0=P0+K01               INCREMENT COUNTER
23BC   6BD1   DMCS 044                                    T1=T1<P1                COMPARE DATA BYTES
23BE   C4D1   DMCS 045                          051  EQUAL      BR IF Z=0          BR IF DATA BYTES EQUAL
23C0   52A2   DMCS 046      NOTLEN                        RDH V DA,AC             RESTORE DATA ADDRESS
23C2   6773   DMCS 047                                    D1=D1+D1                UPDATE PCCL CONSTANT
23C4   D72C   DMCS 048                          036  TRYLEN     BR IF D11=0        BR IF CONSTANT NO LENGTH 240
23C6   21F5   DMCS 049                                    U1=0$KF0                SET CHAIN LENGTH TO 240
23C8   A3D4   DMCS 050                          053  SETHDW     BR                 GO TO SET PCCL FOR LENGTH 240
23D0   0CFD   DMCS 051      EQUAL                         Z=P0<KF0                TEST FOR LAST COMPARE
23D2   C4B6   DMCS 052                          041  COMPAR     BR IF ZNZ          GO BACK IF NOT LAST COMPARE
23D4   497F   DMCS 053      SETHDW                        PCCL=D1                 SET HDWR LATCH TO CHAIN LENGTH
23D6   2C75   DMCS 055                                    P0=0$K70                BUILD XLATE TABLE ADDR FOR NO24K
23D8   2D05   DMCS 059                                    P1=0$K00                SET ADDR TO 7X00 OR 9X00
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS
23DA   55C0    DMCS 060                                   RDB G1 AS,P          READ PCCL CONSTANT
23DC   C560    DMCS 061                 063    RESTG1     BR IF G10=0          BR IF NO BLOCK DATA CHECK BIT
23DE   3785    DMCS 062                                   D1=D1$K80            SET BLOCK DATA CHK BIT FOR REGEN
23E0   2505    DMCS 063    RESTG1                         G1=0$K00             CLEAR G1 REGISTER
23E2   52A2    DMCS 064                                   RDH V DA,AC          RESTORE INITIAL DATA ADDRESS
               DMCS 065    *
               DMCS 066    *                                          CLEAR  MCS  TABLE  ROUTINE
               DMCS 067    *
23E4   28F7    DMCS 068                                   T1=0$KFF             BUILD CLEAR CHARACTER OF FF
23E6   78C8    DMCS 069    CLEAR                          STB T1 AS,P+1        CLEAR TRANSLATE
23E8   F866    DMCS 070                 069    CLEAR      BR IF PC7=0             TABLE TO FF
23EA   2805    DMCS 071                                   T1=0$K00             BUILD 00 IN POS 40 XLATE TBL
23EC   B248    DMCS 072                 081    STZERO     BR                   GO STORE 00 IN POS 40 XLATE TBL
               DMCS 073    *
               DMCS 074    *                                          LOAD  MCS  TABLE  ROUTINE
               DMCS 075    *
323E   5D38    DMCS 076    RDLOOP                         RDB P1 V+1           READ CHAIN IMAGE BYTE
3240   3585    DMCS 077                                   G1=G1$K80            TURN ON NEW CHARACTER SWITCH
3242   281B    DMCS 078                                   T1=T1+K01            INCREMENT PRINT POSITION COUNTER
3244   7BC0    DMCS 079    STLOOP                         STB T1 AS,P          STORE PRINT POS NUMBER IN TABLE
3246   FE4F    DMCS 080                 084    NOFOLD     BR IF HC BIT3=1      BR AROUND FOLD ROUTINE IF ON
3248   2D4D    DMCS 081    STZERO                         P1=P1+K40            PLUS 40 HEX TO ADDRESS FOR FOLD
324A   264D    DMCS 082                                   D0=D0+K40            ADD 40 TO NUMBER OF FOLDS CTR
324C   F4C4    DMCS 083                 079    STLOOP     BR IF DYN BIT3=0     STAY IN LOOP 4 TIMES
324E   CBE7    DMCS 084    NOFOLD       100    CKLAST     BR IF PRD BIT4=1     BRANCH IF NOT DUALING
3250   D767    DMCS 085                 100    CKLAST     BR IF D1 BIT1=1      BR AROUND DUAL RTN IF 240 CHAR
3252   C566    DMCS 086                 100    CKLAST     BR IF G1 BIT0=0      BR IF NEW CHARACTER SWITCH OFF
               DMCS 087    *
               DMCS 088    *                                          DUALING  ROUTINE
               DMCS 089    *
3254   2515    DMCS 090                                   G1=0$K10             SET REG TO DUALED PAIR LIST
3256   C167    DMCS 091    RDLIST       100    CKLAST     BR IF G1 BIT4=1      BR WHEN REG POINTS PAST LIST
3258   4948    DMCS 092                                   RDB I1 CS,G+1        GET ONE OF THE DUALED CHARACTERS
325A   69D1    DMCS 093                                   I1=I1<P1             COMPARE TO CHAIN IMAGE CHARACTER
325C   C4D6    DMCS 094                 091    RDLIST     BR IF ZNZ            BR IF NOT EQUAL
325E   F163    DMCS 095                 097    ADDROK     BR IF G1 BIT7=1      BR IF MATCH WAS FIRST OF PAIR
3260   6446    DMCS 096                                   G=G-2                SET ADDR TO CHAR TO BE DUALED
3262   4D40    DMCS 097    ADDROK                         RDB P1 CS,G          GET TABLE ADDR OF DUALED CHAR
3264   B244    DMCS 098                 079    STLOOP     BR                   GO STORE PRINT POSITION NUMBER
               DMCS 099    *                                                      IN DUAL CHARACTER POSITION
3266   5B59    DMCS 100    CKLAST                         G1=T1                TEST FOR
3268   6511    DMCS 101                                   G1=G1<U1                 LAST CHARACTER
326A   C4BE    DMCS 102                 076    RDLOOP     BR IF ZNZ            BRANCH IF NOT LAST CHARACTER
               DMCS 103    *
               DMCS 104    *                                          END HOUSEKEEPING
               DMCS 105    *
326C   2D05    DMCS 106                                   P1=0$K00             SET ADDR TO 7000 & 9000
326E   77C0    DMCS 107                                   STB D1 AS,P          STORE PCCL CONSTANT IN TABLE
3270   54E2    DMCS 108                                   RDH G DA,BC          RESTORE G REG
3272   56F2    DMCS 109                                   RDH D DA,BE          RESTORE D REG
3274   2105    DMCS 110                                   U1=0$K00             ZERO COUNT
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

3276     2E43     DMCS 111                                    HO=O$KC4               SET DEVICE END
3278     8D48     DMCS 112                DPTR 055  CHEND     BR                     GO TO CHANNEL END ROUTINE
                  DMCS 113   *
                  DMCS 114   ATABLE  ADDR=220A
220A     1028     DMCS 115   C                                 XCTL'1028303C50784C5D6C4D7C7D787E'
                  DMCS 116   AEND
                                        ************************************
                                        * CROSS REFERENCE FOR CSECT DMCS *
                                        ************************************

DMCS 003     DPTS 077
DMCS 009     DMCS 006
DMCS 018     DMCS 016
DMCS 020     DPTS 093
DMCS 022     DMCS 004  DMCS 005  DMCS 007  DMCS 023
DMCS 023     DMCS 020
DMCS 026     DMCS 021
DMCS 036     DMCS 048
DMCS 041     DMCS 052
DMCS 051     DMCS 045
DMCS 053     DMCS 050
DMCS 063     DMCS 061
DMCS 069     DMCS 070
DMCS 076     DMCS 102
DMCS 079     DMCS 083  DMCS 098
DMCS 081     DMCS 072
DMCS 084     DMCS 080
DMCS 091     DMCS 094
DMCS 097     DMCS 095
DMCS 100     DMCS 084  DMCS 085  DMCS 086  DMCS 091
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS
                DPTC 001    T                          1403 PRINT BUFFER SCAN ROUTINE
                DPTC 002    *
                DPTC 003    *          THIS ROUTINE CAN BE USED TO DISPLAY THE CONTENTS OF THE 1403 PRINT
                DPTC 004    *          LINE BUFFER (PLB). THE PLB CONSISTS OF 8 DATA PLANES PLUS A PARITY
                DPTC 005    *          PLANE. IN ADDITION THERE ARE 4 CHECK PLANES WHICH ARE USED TO
                DPTC 006    *          INDICATE THE CAUSE OF A HAMMER CHECK AT THE END OF A PRINT OPERA-
                DPTC 007    *          TION. THIS ROUTINE WILL PRINT A 4-CHARACTER GROUP ON THE 1052 FOR
                DPTC 008    *          EACH PRINT POSITION IN PRINT POSITION SEQUENCE. A SPACE WILL OCCUR
                DPTC 009    *          BETWEEN EACH GROUP. AT THE END OF 16 GROUPS A NEW LINE WILL BE
                DPTC 010    *          STARTED. THE PRINT-OUT WILL CONTINUE UNTIL 132 GROUPS HAVE BEEN
                DPTC 011    *          PRINTED. THIS ROUTINE IS ENTERED BY USING THE ALTER/DISPLAY KEY.
                DPTC 012    *          'DB' IS THEN TYPED ON THE 1052. THE 132 GROUPS ARE PRINTED OUT
                DPTC 013    *          WITHOUT ANY FURTHER OPERATIONS. 'EOB' AND 'CANCEL' ARE INEFFECTIVE
                DPTC 014    *          DURING THE PRINT-OUT. AT THE END OF THE PRINT-OUT THE OPERATION
                DPTC 015    *          RETURNS TO THE ALTEP/DISPLAY ROUTINE FOR ADDITIONAL ALTER/DISPLAY
                DPTC 016    *          OPERATIONS OR FOR RETURN TO CPU CONTROL.
                DPTC 017    *
                DPTC 018    *          THE FIRST CHARACTER OF EACH GROUP IS THE CONTENTS OF THE CHECK
                DPTC 019    *          PLANES. LISTED BELOW ARE THE CHARACTERS WHICH MAY BE PRINTED AND
                DPTC 020    *          THE CONDITIONS THEY REPRESENT. AHDO MEANS ADDRESSED HD OFF.
                DPTC 021    *
                DPTC 022    *
                DPTC 023    *                0,6       MAY OCCUR IF A COIL PROTECT CHECK TERMINATED
                DPTC 024    *                          PRINTING                              ***NO ERROR***
                DPTC 025    *                1,2,3     SHOULD NOT OCCUR
                DPTC 026    *                4         MAY OCCUR ON A 1403-N1 IF A COIL PROTECT CHECK
                DPTC 027    *                          TERMINATED PRINTING                   ***NO ERROR***
                DPTC 028    *                5,7,C,D   SHOULD NOT OCCUR
                DPTC 029    *                8         PRINTED PROPERLY                      ***NO ERROR***
                DPTC 030    *                9         NO COMPARE OCCURRED AFTER SUFFICIENT PRINT SCANS
                DPTC 031    *                          HAD TAKEN PLACE                       ***ERROR***
                DPTC 032    *                A         THE AHDO CIRCUIT INDICATED THAT A HAMMER DRIVER
                DPTC 033    *                          WAS ON, BUT NO COMPARE HAD OCCURRED.  ***ERROR***
                DPTC 034    *                B         THE AHDO CIRCUIT INDICATED THAT A HAMMER DRIVER
                DPTC 035    *                          WAS ON, BUT BUT A PLC BIT WAS ON DUE TO A
                DPTC 036    *                          BLANK OR PREVIOUS PRINTING            ***ERROR***
                DPTC 037    *                E         A COMPARE OCCURRED, BUT THE AHDO INDICATED
                DPTC 038    *                          THAT THE HAMMER DRIVER DID NOT TURN ON.  ***ERROR***
                DPTC 039    *                F         A COMPARE OCCURRED, BUT A PLC BIT WAS ON DUE TO
                DPTC 040    *                          A BLANK OR PREVIOUS PRINTING.         ***ERROR***
                DPTC 041    *
                DPTC 042    *          IF A COIL PROTECT CHECK OCCURRED DURING PRINTIN, AN 'A' OR 'B'
                DPTC 043    *          CODE SHOULD BE PRESENT FOR THE POSITION CAUSING THE CHECK.
                DPTC 044    *
                DPTC 045    *          THE SECOND CHARACTER OF EACH GROUP IS USED TO INDICATE THE PARITY
                DPTC 046    *          CONDITION OF THE DATA PLANES. A '1' INDICATES A PLB PARITY CHECK.
                DPTC 047    *
                DPTC 048    *          THE THIRD AND FOURTH CHARACTERS OF EACH GROUP DISPLAY THE PRINT DATA
                DPTC 049    *          IN THE DATA PLANES OF THE PLB AS FOLLOWS.
                DPTC 050    *
                DPTC 051    *                  0  PLB 128          4  PLB  8
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

                DPTC 052    *                      1  PLB  64        5  PLB  4
                DPTC 053    *                      2  PLB  32        6  PLB  2
                DPTC 054    *                      3  PLB  16        7  PLB  1
                DPTC 055    *
                DPTC 056    *          THE DATA LOADED IN THE PLB WAS OBTAINED AS A RESULT OF USING THE
                DPTC 057    *          1403 TRANSLATE TABLE IN AUXILIARY STORAGE.
                DPTC 058    *        ************************************************************************
2F78    2185    DPTC 059     START                        U1=0$K80           SET ADDRESS CONSTANT TO
2F7A    2138    DPTC 060                                  U1=U1+K03               HEX 83 (DECIMAL 131)
2F7C    5404    DPTC 061                                  G=U+1              CHARACTER COUNT G1=84 (DEC 132)
2F7E    2413    DPTC 062                                  G0=0$KC1           SET PLBAR COUNT TO 01
2F80    88BC    DPTC 063     FEED       ALDP 241  LINEFD  BAL                FORCE A LINE FEED
2F82    2482    DPTC 064     TOP                          SET MODE K=18      SET 1403 MODE
2F84    4F4F    DPTC 065                                  PR=G0              SET PLBAR
2F86    1D00    DPTC 066                                  RST PRA K=80       RESET THE PARITY CHECK LATCH
2F88    2D04    DPTC 067                                  SET PRA K=20       SET READ CONTROL
2F8A    2F10    DPTC 068                                  SET PRB K=01       SET DIAG DECODE 4
2F8C    582F    DPTC 069                                  V0=PRI             READ BUFFER DATA REG
2F8E    5F0F    DPTC 070                                  U0=PRD             READ DIAG BYTE
2F90    10F3    DPTC 071                                  U0=U0*-K0F         CLEAR LOWER DIGIT OF CHECK PLANE
2F92    EA96    DPTC 072                    074  NPCHK    BR IF PRS BIT6=0   TEST FOR PARITY CHECK
2F94    3013    DPTC 073                                  U0=U0$K01          SET PARITY CHECK BIT
2F96    2404    DPTC 074     NPCHK                        SET MODE K=20      SET 1052 MODE
2F98    2F43    DPTC 075                                  H1=0$KC4           RESET DIGIT COUNTER
2F9A    5035    DPTC 076     NEXT                         V1=U0XL            PREPARE TO UNPACK UPPER DIGIT
2F9C    23FD    DPTC 077     UNPACK                       V1=V1+KF0          UNPACK HEX TO EBCDIC, ADD F0
2F9E    0361    DPTC 078                                  Z=V1+K06           TEST FOR DIGIT HIGHER THAN 9
2FA0    F4A4    DPTC 079                    081  PRINT    BR IF AC=0         BR IF DIGIT IS 0 THROUGH 9
2FA2    73B1    DPTC 080                                  V1=V1-T1+1         T1=39 GET C1 THRU C6
2FA4    88CE    DPTC 081     PRINT      ALDP 252  TYPCHR  BAL                GO TO TYPE CHARACTER ON 1052
2FA6    503D    DPTC 082                                  V1=U0L             PREPARE TO UNPACK LOWER DIGIT
2FA8    2FFF    DPTC 083                                  H1=H1+KFF          DECREMENT DIGIT COUNTER
2FAA    FB1D    DPTC 084                    077  UNPACK   BR IF H17=1        BRANCH TO UNPACK LOWER DIGIT
2FAC    5209    DPTC 085                                  U0=V0              MOVE DATA BYTE TO PRINT REGISTER
2FAE    EB1B    DPTC 086                    076  NEXT     BR IF H16=1        BR TO PRINT DATA CHARACTER
2FB0    88CC    DPTC 087                ALDP 251  SPACE   BAL                GO TO SPACE BETWEEN CHAR GROUPS
2FB2    247B    DPTC 088                                  G0=G0+K07
2FB4    24AD    DPTC 089                                  G0=GC+KA0          ADD DECIMAL 167 TO PLBAR COUNT
2FB6    F48B    DPTC 090                    092  OVER88   BR IF AC=1         BR IF PLBAR WAS OVER 88
2FB8    6413    DPTC 091                                  G0=G0+U1           ADD DECIMAL 131 TO PLBAR COUNT
2FBA    242B    DPTC 092     OVER88                       G0=G0+K02          ADD 2 TO PLBAR COUNT
2FBC    25FF    DPTC 093                                  G1=G1+KFF          DECREMENT CHARACTER COUNT
2FBE    C4C2    DPTC 094                    096  NOTEND   BR IF ZNZ          CONTINUE FOR 132 POSITIONS
2FC0    A6AE    DPTC 095                ALDP 302  RSTLCH  BR                 DONE-RETURN TO ALTER/DISPLAY
2FC2    054B    DPTC 096     NOTEND                       Z=G1<K04           TEST FOR X4 PLBAR COUNT
2FC4    F082    DPTC 097                    064  TOP      BR IF LZNZ         BR IF NOT X4-NOT END OF LINE
2FC6    AF80    DPTC 098                    063  FEED     BR                 END OF LINE- CAUSE A LINE FEED
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT DPTC *
                                  **********************************

DPTC  059      ALDP  103
DPTC  063      DPTC  098
DPTC  064      DPTC  097
DPTC  074      DPTC  072
DPTC  076      DPTC  086
DPTC  077      DPTC  084
DPTC  081      DPTC  079
DPTC  092      DPTC  090
DPTC  096      DPTC  094
```

DPTQ DESCRIPTIVE TEXT

ENTRY POINTS

PTRREQ - ENTRY IS MADE AT THIS LABEL FROM THE -BSWI- ROUTINE. BIT 7 OF THE EXTERNAL (PRS) HAD BEEN TESTED AND FOUND TO BE ON. THIS INDICATES THAT A PRINT REQUEST HAS BEEN GENERATED.

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

            DPTQ 001    T                              1403 INTERRUPT REQUEST ROUTINE
            DPTQ 002    *          ********************************************************************
30C4  4642  DPTQ 003     PTRREQ                        RDH D DC,98          D=00C8---1403 UCW ADDRESS
30C6  8F74  DPTQ 004               ERRQ 079  RDCONT    BAL    .             GET CH AND UNIT STAT AND COUNT
30C8  D158  DPTQ 005                    C13  NACTIV     BR IF G15=0          BR IF NO ACTIVE FLAG
30CA  C468  DPTQ 006                    021  IFCHEK     BR IF G00=0          BR IF NO SECONDARY
30CC  CED4  DPTQ 007                    011  CHAINC     BR IF PRS0=0         BR IF NO EXT DEVICE END
30CE  8F00  DPTQ 008     CH12      DPTT 071  SENCHK     BAL                  GO TO DEVICE END SENSE
30D0  0EC3  DPTQ 009                                    Z=H0*-K0C            LOOK FOR UNIT CHECK OR
30D2  F0E0  DPTQ 010                    017  RSTCHN     BR IF LZNZ           UNIT EXCEPTION
30D4  D562  DPTQ 011     CHAINC         018  TAKEIN     BR IF G11=0          BR IF NO CMD CHAINING
30D6  8D64  DPTQ 012               DPTR 073  CHAIOK     BR                   GO TO CMD CHAIN SET UP
30D8  C465  DPTQ 013     NACTIV         019  DEVEN1     BR IF G00=1          BR IF SECONDARY BIT
30DA  CAE8  DPTQ 014                    021  IFCHEK     BR IF PRS4=0         BR IF NO INITIAL READY
30DC  3485  DPTQ 015                                    G0=G0$K80            SET SECONDARY BIT
30DE  2E43  DPTQ 016                                    H0=0$K04             SET DEVICE END
30E0  15C5  DPTQ 017     RSTCHN                         G1=G1*-KC0           RESET CHAINING
30E2  8D68  DPTQ 018     TAKEIN    DPTR 075  TAKEIN     BR
30E4  CECF  DPTQ 019     DEVEN1         008  CH12       BR IF PRS0=1         BR IF EXT DEVICE END
30E6  DA63  DPTQ 020                    018  TAKEIN     BR IF H05=1          BR IF DEVICE END IN UCW
30E8  AB20  DPTQ 021     IFCHEK    BSWI 107  RESTRH     BR                   RETURN TO NATIVE REQUEST
                                   **********************************
                                   * CROSS REFERENCE FOR CSECT DPTQ *
                                   **********************************

DPTQ 003    BSWI 103
DPTQ 008    DPTQ 019
DPTQ 011    DPTQ 007
DPTQ 013    DPTQ 005
DPTQ 017    DPTQ 010
DPTQ 018    DPTQ 011  DPTQ 020
DPTQ 019    DPTQ 013
DPTQ 021    DPTQ 006  DPTQ 014
```

DPTR DESCRIPTIVE TEXT

ENTRY POINTS

WRITE - ENTRY IS MADE AT THIS LABEL FROM THE -DPTS- ROUTINE WHEN A 1403 WRITE COMMAND HAS BEEN DECODED. THIS ROUTINE LOADS DATA FROM THE CPU INTO THE PRINT LINE BUFFER. WHEN THE PLB COUNT AND THE UCW COUNT BOTH REACH ZERO, CHANNEL END IS SET.

WXFER3 - ENTRY IS MADE AT THIS LABEL FROM THE -DPTS- ROUTINE WHEN DATA CHAINING IS IN EFFECT.

CHEND - ENTRY IS MADE AT THIS LABEL FROM THE -DMCS- ROUTINE TO SET CHANNEL END AT THE COMPLETION OF AN MCS LOAD OPERATION.

TSTCDA - ENTRY IS MADE AT THIS LABEL FROM THE -DPTS- ROUTINE TO TEST FOR CHAIN DATA AND THE SLI FLAG AND TO SET CHANNEL END FOR A SENSE COMMAND. ENTRY CAN ALSO BE MADE FROM THE -DYPE-, -EEDR- AND -ERDR- ROUTINES SINCE THEY ALSO ENTER THE -DPTS- SENSE ROUTINE.

ENDOK - ENTRY IS MADE AT THIS LABEL FROM THE -DPTS- ROUTINE TO SET CHANNEL END IF THE UCW COUNT WAS ONE.

CCBR - ENTRY IS MADE AT THIS LABEL FROM THE -EEDR- AND -EXFR- ROUTINES TO TEST FOR CHAINING AND TO TAKE A CHANNEL END INTERRUPT.

CHAINC - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE- ROUTINE TO RESTORE THE P AND I REGISTERS, UPDATE THE UCW AND GO TO CHAINING.

CHAIOK - ENTRY IS MADE AT THIS LABEL FROM THE -DPTQ-, -EERQ- AND -ERRQ- ROUTINES TO UPDATE THE UCW AND GO TO CHAINING.

TAKEIN - ENTRY IS MADE AT THIS LABEL FROM THE -BWRP-, -DCLR-, -DPTQ-, -DPTS-, -EERQ-, AND -ERRQ- ROUTINES TO ATTEMPT TO TAKE AN INTERRUPT.

STINST - ENTRY IS MADE AT THIS LABEL FROM THE -EEDR-, -ERDR- AND -EXFR- ROUTINES TO UPDATE THE UCW.

IMMEDI - ENTRY IS MADE AT THIS LABEL FROM THE -DPTS-, -EEDR- AND -EXFR- ROUTINES TO PROVIDE CHANNEL END FOR AN IMMEDIATE COMMAND.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

             DPTR 001    T                                 1403 WRITE-CHANNEL END,AND CARRIAGE CTRL ROUTINE
             DPTR 003    *          ***************************************************************
             DPTR 004    *                                  ***   WRITE COMMANDS ***
             DPTR 005    *          ***************************************************************
             DPTR 006    ASEQ    AL07=70
0CF0   E874  DPTR 007     WRITE          009   NOLOAD     BR IF H16=0            BR IF NOT GATE LOAD TAG
0CF2   2D20  DPTR 008                                     SET PRA K=02           SET BLOCK 2 HOME PULSES LATCH
0CF4   2813  DPTR 009     NOLOAD                          IC=0$K01               SET PLBAR COUNT 01
0CF6   2985  DPTR 010                                     I1=0$K80
0CF8   293B  DPTR 011                                     I1=I1+K03              SET ADDRESS CONSTANT TO 131
0CFA   6A99  DPTR 012                                     TOC=T0+I1+1            TO WAS 0 SET PLBAR COUNT TO 132
0CFC   2C75  DPTR 014                                     P0=0$K70               SET PLB TABLE ADDR FOR NOT 24K
0CFE   5D38  DPTR 018     WXFER1                          RDB P1 V+1             READ CHAR FROM DATA ADDR
0D00   5DB9  DPTR 019                                     T1=P1                  MOVE POSSIBLE 00 TO T1
0D02   C48D  DPTR 020                 025   WXFER2      BR IF Z=0              TEST IF P1 WAS 00
0D04   5BC0  DPTR 021                                     RDB T1 AS,P+0          READ CHAR FROM TABLE
0D06   CBFF  DPTR 022                                     Z=T1<KFF
0D08   C48C  DPTR 023                 025   WXFER2      BR IF ZNZ              BRANCH IF CHAR NOT FF
0D0A   3F83  DPTR 024                                     H1=H1$K08              SET DATA CHECK IN UNIT SENSE
0D0C   4F8F  DPTR 025     WXFER2                          PR=IO                  LOAD PLBAR
0D0E   2D04  DPTR 026                                     SET PRA K=20           SET READ CONTROL
0D10   287B  DPTR 027                                     IO=IO+K07              ADD7 TO PLBAR COUNT--DELAY LOAD
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS
0D12  4BBF  DPTR 028                                   PRO=T1               LOAD PLB
0D14  28AD  DPTR 029                                   IO=IO+KA0            ADD DECIMAL 160 TO PLBAR COUNT
0D16  F49B  DPTR 030                 032  OVER88      BR IF AC=1           BR IF PLBAR WAS OVER 88
0D18  6893  DPTR 031                                   IO=IO+I1             ADD DECIMAL 131 TO PLBAR COUNT
0D1A  282B  DPTR 032  OVER88                           IO=IO+K02            ADD 2 TO PLBAR COUNT
0D1C  D42B  DPTR 033                 040  XFER00      BR IF G01=1          BR IF INC LENGTH LOAD ZEROS
0D1E  5006  DPTR 034                                   U=U-1                UCW COUNT MINUS 1
0D20  C4A9  DPTR 035                 039  COUNT0      BR IF Z=0            BR IF UCW COUNT IS 0
0D22  2AFF  DPTR 036  WXFER3                           T0=T0+KFF            PLB COUNT MINUS 1
0D24  C4B5  DPTR 037                 045  WRGLEN      BR IF Z=0            BR IF PLB COUNT IS 0
0D26  8CFE  DPTR 038                 018  WXFER1      BR                   GO TO NEXT BYTE
0D28  C565  DPTR 039  COUNT0         073  CHAIOK      BR IF G10=1          BR IF DATA CHAIN FLAG
0D2A  2AFF  DPTR 040  XFER00                           T0=T0+KFF            PLB COUNT MINUS 1
0D2C  C4B7  DPTR 041                 046  XFREND      BR IF Z=0            BR IF PLB COUNT IS 0
0D2E  3445  DPTR 042                                   G0=G0$K40            SET INCORRECT LENGTH
0D30  2B05  DPTR 043                                   T1=0$K00             SET PLB BYTE TO 00
0D32  8D0C  DPTR 044                 025  WXFER2      BR                   GO TO NEXT LOAD 00 BYTE
0D34  3445  DPTR 045  WRGLEN                           G0=G0$K40            SET INCORRECT LENGTH
0D36  EABE  DPTR 046  XFREND         050  NPCHK1      BR IF PRS BIT6=0     BR IF NO PARITY CHECK
0D38  3E23  DPTR 047                                   H0=HC$K02            SET UNIT CHECK STATUS
0D3A  3F15  DPTR 048                                   H1=H1$K10            SET EQUIP CHECK SENSE
0D3C  1545  DPTR 049                                   G1=G1*-K40           RESET COMMAND CHAIN FLAG
0D3E  2D05  DPTR 050  NPCHK1                           P1=0$KC0             SET ADDR TO 7X00 OR 9X00
0D40  5DC0  DPTR 051                                   RDB P1 AS,P          READ PCCL CONSTANT
0D42  CD46  DPTR 052                 054  CHKEND      BR IF P10=0          BR IF NO BLOCK DATA CHECK BIT
0D44  1F83  DPTR 053                                   H1=H1*-K08           RESET DATA CHECK
0D46  3D02  DPTR 054   CHKEND                          SET PRA K=90         SET PRINT GATE AND BUSY
0D48  1F23  DPTR 055   CHEND                           H1=H1*-K02           RESET GATE LOAD TAG
0D4A  D456  DPTR 056                 066  ENDOK       BR IF G01=0          BR IF NO INCORRECT LENGTH
            DPTR 058  *              ********************************************************************************
            DPTR 059  *                    CHANNEL END ROUTINE
            DPTR 060  *              ********************************************************************************
0D4C  C555  DPTR 061   TSTCDA        065  NOSLI       BR IF G10=1          BR IF DATA CHAIN FLAG
0D4E  E554  DPTR 062   INCLEN        065  NOSLI       BR IF G12=0          BR IF NO SLI FLAG
0D50  1445  DPTR 063                                   G0=G0*-K40           RESET INCORRECT LENGTH
0D52  8D56  DPTR 064                 066  ENDOK       BR
0D54  1545  DPTR 065   NOSLI                           G1=G1*-K40           RESET COMMAND CHAIN FLAG
0D56  3E83  DPTR 066   ENDOK                           H0=HC$K08            SET CHANNEL END
0D58  3485  DPTR 067                                   G0=G0$K80            SET SECONDARY BIT
0D5A  D568  DPTR 068   CCBR          075  TAKEIN      BR IF G11=0          BR IF NO CHAIN--TRY CH END INTRP
0D5C  8FBA  DPTR 069   CMDCHN  DCOM 006  PCIBR       BAL                  CHECK FOR PCI INTERRUPT
0D5E  1E83  DPTR 070                                   H0=HC*-K08           RESET CHAN END--FOR CHAINING
0D60  DA6A  DPTR 071                 076  STINST      BR IF H05=0          BR IF NO DEVICE END YET
0D62  8178  DPTR 072   CHAINO  CCOM 066  LRSTRB      BAL                  RESTORE P AND I REGS
0D64  B404  DPTR 073   CHAIOK  DCOM 080  CHANOK      BAL                  UPDATE UCW
0D66  95CA  DPTR 074           DCLB 043  CHAIN       BR                   GO TO CHAIN
0D68  8FBE  DPTR 075   TAKEIN  DCOM 008  INTRPT      BAL                  TRY NON-PCI INTERRUPT
0D6A  B406  DPTR 076   STINST  DCOM 081  REDSTR      BAL                  UPDATE UCW
0D6C  AB20  DPTR 077           BSWI 107  RESTRH      BR                   SET CPU MODE-GO TO NATV REQ
            DPTR 079  *              ********************************************************************************
            DPTR 080  *                    IMMEDIATE COMMAND CHANNEL END ROUTINE
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

                DPTR 081    *          ********************************************************************
0D6E     C572   DPTR 082     IMMEDI      084     OKCHAI     BR IF G10=0          BR IF NOT CHAIN DATA
0D70     15E5   DPTR 083                                    G1=G1*-KE0           RESET CC,CD,SLI FLAGS
0D72     3485   DPTR 084     OKCHAI                         G0=G0$K80            SET SECONDARY BIT
0D74     3E83   DPTR 085                                    H0=H0$K08            SET CHANNEL END
0D76     D55D   DPTR 086                 069     CMDCHN     BR IF G11=1          BR IF THIS COMMAND CHAINED
0D78     F269   DPTR 087                 075     TAKEIN     BR IF D07=1          BR IF LAST COMMAND CHAINED
0D7A     DA7E   DPTR 088                 090     NOTNOP     BR IF H05=0          BR IF NO DEVICE END
0D7C     1485   DPTR 089                                    G0=G0*-K80           RESET SECONDARY BIT
0D7E     8C60   DPTR 090     NOTNOP   DPTS 137   PCENTR     BR
                DPTR 091    AEND

                                     ***********************************
                                     * CROSS REFERENCE FOR CSECT DPTR *
                                     ***********************************

DPTR 007    DPTS 090
DPTR 009    DPTR 007
DPTR 018    DPTR 038
DPTR 025    DPTR 020  DPTR 023  DPTR 044
DPTR 032    DPTR 030
DPTR 036    DPTS 047
DPTR 039    DPTR 035
DPTR 040    DPTR 033
DPTR 045    DPTR 037
DPTR 046    DPTR 041
DPTR 050    DPTR 046
DPTR 054    DPTR 052
DPTR 055    DMCS 112
DPTR 061    DPTS 114
DPTR 065    DPTR 061  DPTR 062
DPTR 066    DPTR 056  DPTR 064  DPTS 115
DPTR 068    EXFR 144
DPTR 069    DPTR 086
DPTR 072    DYPE 162
DPTR 073    DPTQ 012  DPTR 039  ERRQ 039
DPTR 075    BWRP 152  DCLR 053  DPTQ 018  DPTR 068  DPTR 087  DPTS 140  ERRQ 063
DPTR 076    DPTR 071  ERDR 073  EXFR 142
DPTR 082    DPTS 099  DPTS 126  EXFR 131
DPTR 084    DPTR 082
DPTR 090    DPTR 088
```

DPTS DESCRIPTIVE TEXT

ENTRY POINTS

PRSIO - ENTRY IS MADE AT THIS LABEL FROM THE -DCLB- ROUTINE. A START I/O WAS DETECTED AND THE CCW HAS BEEN FETCHED.

SENSE - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE- ROUTINE TO HANDLE A SENSE COMMAND FOR THE 1052. ENTRY IS ALSO MADE TO THE -DYPE- SENSE ROUTINE FROM THE -EEDR- AND -ERDR- ROUTINES.

NOOP - ENTRY IS MADE AT THIS LABEL FROM THE -DMCS-, -EEDR- AND -ERDR- ROUTINES TO SET CHANNEL END AND DEVICE END FOR NO-OP OR NO-OP TYPE COMMANDS.

PCENTR - ENTRY IS MADE TO THIS LABEL FROM THE -DPTR- ROUTINE TO UPDATE THE UCW FOR AN IMMEDIATE COMMAND. ENTRY IS ALSO MADE TO THIS LABEL FROM THE -EEDR- AND -ERDR- ROUTINES TO UPDATE THE UCW WHEN A CMD REJECT IS DETECTED.

TSTPCI - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE- ROUTINE TO HANDLE STATUS FOR A 1052 NO-OP OR NOT READY CONDITION.

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS

          DPTS 001    T                          1403 PRINTER START I/O, COMMAND DECODE AND SENSE
          DPTS 003    *              1403 START IO ACTIVE 0 SECONDARY 0
          DPTS 004    *                                        COUNT            U
          DPTS 005    *                                        DATA ADDRESS     V
          DPTS 006    *                                        COMMAND BYTE     G0
          DPTS 007    *                                        FLAGS AND OP     G1
          DPTS 008    *                                        CHANNEL STATUS   H0
          DPTS 009    *                                        UCW ADDRESS D    OXC8
          DPTS 010    *                                        UCW STATUS BYTE  OXCA
          DPTS 011    *                                        UCW SENSE BYTE   OXCB
          DPTS 012    *
          DPTS 013    *                      ****************************************
          DPTS 014    *                      NATIVE 1403 UCW FORMAT
          DPTS 015    *                      ****************************************
          DPTS 016    *
          DPTS 017    *       C8                  CA                  CC                  CE
          DPTS 018    *      *-------------------------------------------------------------------*
          DPTS 019    *      *        *         *         *         *                     *                *
          DPTS 020    *      * CHANNEL* CHANNEL*  UNIT   *  UNIT   *     RESIDUAL        *   NEXT  CCW    *
          DPTS 021    *      *  STATUS* FLAGS  * STATUS  * SENSE   *      COUNT          *    ADDRESS     *
          DPTS 022    *      *        *         *         *         *                     *                *
          DPTS 023    *      *-------------------------------------------------------------------*
          DPTS 024    *
          DPTS 025    *
          DPTS 026    *
          DPTS 027    *      CH STAT-C8        CHANNEL FLAGS-C9     UNIT STAT-CA       UNIT SENSE-CB
          DPTS 028    *      0-SEC             0-CHAIN DATA         0-                 0-COMM REJ
          DPTS 029    *      1-INC LENGTH      1-COMM CHAIN         1-                 1-INTERV REQ
          DPTS 030    *      2-PROG CHECK      2-SLI                2-                 2-
          DPTS 031    *      3-PROT CHECK      3-SKIP               3-BUSY             3-EQUIP CHK
          DPTS 032    *      4-CH DATA CHK     4-PCI                4-CH END           4-DATA CHK
          DPTS 033    *      5-CH CTRL CHK     5-ACTIVE             5-DEV END          5-
          DPTS 034    *      6-                6-INPUT              6-UNIT CHK         6-GATE LOAD
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT            COMMENTS

                DPTS 035    *              7-INT IN BUFFER  7-OUTPUT             7-UNIT EXEC       7-CHAN 9
                DPTS 036    *
                DPTS 037    *                               ****************************************
                DPTS 038    *                               START I/O ACTIVE-0 SECONDARY-0
                DPTS 039    *                               ****************************************
                DPTS 040    *
                DPTS 041    *
                DPTS 042    ASEQ    ALO7=72
0BF2    2482    DPTS 043     PRSIO                          SET MODE K=18        SET 1403 MODE
0BF4    5449    DPTS 044                                    GO=GO
0BF6    C4FC    DPTS 045                  048  NCHDAT      BR IF ZNZ            BR IF NO DATA CHAIN
0BF8    5460    DPTS 046                                    RDH G AS,D+0         GET CHANNEL STATUS AND FLAGS
0BFA    8D22    DPTS 047             DPTR 036  WXFER3      BR                   GO TO DATA CHAIN RETURN
0BFC    9816    DPTS 048     NCHDAT  CCOM 058  LSAVEB      BAL                  SAVE P AND I REGS
0BFE    54C9    DPTS 049                                    P0=G0                MOVE COMMAND TO P0
0C00    8F7A    DPTS 050             ERRQ 082  RDCHST      BAL                  CHANNEL STATUS AND FLAGS G0 G1
0C02    DE87    DPTS 051                  053  READY       BR IF PRS BIT1=1     BR IF PRINT READY ON
0C04    3F45    DPTS 052                                    H1=H1$K40            SET INTERVENTION REQUIRED
0C06    0C4B    DPTS 053     READY                          Z=P0<K04             UNIT STATUS AND SENSE H0 H1
0C08    C4BF    DPTS 054                  104  SENSE       BR IF Z=0            BRANCH IF SENSE COMMAND
0C0A    0C3B    DPTS 055                                    Z=P0<K03
0C0C    C4D7    DPTS 056                  121  NOOPTR      BR IF Z=0            BR IF NOOP TO TEST FOR INTV REQD
                DPTS 057    *              ****************************************************************
                DPTS 058    *                    DECODE REMAINDER OF THE COMMANDS
                DPTS 059    *              ****************************************************************
0C0E    DE92    DPTS 060                  062  NOTRDY      BR IF PRS BIT1=0     BRANCH IF PRINT READY IS OFF
0C10    1F45    DPTS 061                                    H1=H1*-K40           CLEAR INTERVENTION REQUIRED
0C12    1D0C    DPTS 062     NOTRDY                         RST PRA K=80         RESET EXTERNAL SENSE LATCHES
0C14    1F97    DPTS 063                                    H1=H1*-K99           REGEN INTV REQD AND GATE LOAD
0C16    DF5F    DPTS 064                  132  ISLCHK      BR IF H1 BIT1=1      BR IF INTERVENTION REQUIRED
0C18    D85D    DPTS 065                  131  CMDREJ      BR IF P05=1          REJECT X4,X5,X6,X7,XC,XD,XE,XF
0C1A    F85C    DPTS 066                  131  CMDREJ      BR IF P07=0          REJECT X0,X2,X8,XA
                DPTS 067    *              ****************************************************************
                DPTS 068    *                    TEST FOR VALID COMMANDS AMONG X1,X3,X9 OR XB
                DPTS 069    *              ****************************************************************
0C1C    5CE9    DPTS 070                                    H0=P0                MOVE COMMAND TO H0
0C1E    CC27    DPTS 071                  079  SKIP        BR IF P00=1          BR IF SKIP COMMAND
0C20    0EE9    DPTS 072                                    Z=H0+KE0             TEST FOR SPACE 0,1,2 OR 3
0C22    F4AE    DPTS 073                  087  VALCMD      BR IF AC=0           BR IF VALID SPACE COMMAND
0C24    A37E    DPTS 077             DMCS 003  CD737B      BR                   BR TO TEST FOR MCS 73 OR 7B
0C26    6EE3    DPTS 079     SKIP                           H0=H0+H0             SHIFT CHANNEL DECODE
0C28    E0DD    DPTS 080                  131  CMDREJ      BR IF HZ=0           BR IF SKIP TO CHANNEL 0
0C2A    0E39    DPTS 081                                    Z=H0+K30             TEST FOR SKIP ABOVE 12
0C2C    F4B7    DPTS 085                  092  TSTMCS      BR IF AC=1           BR TO TEST FOR MCS EB,F3 OR FB
0C2E    2E05    DPTS 087     VALCMD                         H0=0$K00             CLEAR UNIT STATUS
0C30    4DCF    DPTS 088                                    PRC=P0               LOAD CARRIAGE CONTROL REGISTER
0C32    E83B    DPTS 089                  098  CARCTL      BR IF P06=1          TEST FOR WRITE OR CARR CTRL CMD
0C34    8CFC    DPTS 090             DPTR 007  WRITE       BR
0C36    E85C    DPTS 092     TSTMCS          131  CMDREJ      BR IF P06=0          CMD REJECT ON E9,F1,F9
0C38    A398    DPTS 093             DMCS 020  EBF3FB      BR                   GO TO GATE LOAD OR LOAD CMDS
                DPTS 095    *              ****************************************************************
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              DPTS 096    *                    CARRIAGE CONTROL COMMAND
              DPTS 097    *                 **********************************************************************
0C3A    2D02  DPTS 098     CARCTL                          SET PRA K=10        SET PRINT BUSY LATCH
0C3C    8D6E  DPTS 099                     DPTR 082  IMMEDI    BR
              DPTS 101    *                 **********************************************************************
              DPTS 102    *                    SENSE COMMAND
              DPTS 103    *                 **********************************************************************
0C3E    F549  DPTS 104     SENSE              109  SNSRTN     BR IF G13=1         BR IF SKIP FLAG ON
0C40    5F89  DPTS 105                                         T1=H1               MOVE SENSE BYTE
0C42    E247  DPTS 106                        108  OVER       BR IF D06=1         SKIP NEXT WORD IF 2540 SHARE
0C44    1B23  DPTS 107                                         T1=T1*-K02          STRIP SENSE BYTE FOR 1403
0C46    7B38  DPTS 108     OVER                               STB T1 V+1          STORE SENSE BYTE
0C48    2E43  DPTS 109     SNSRTN                             H0=0$K04            CLEAR UNIT STATUS SET DEVICE END
0C4A    5006  DPTS 110                                         U=U-1               UCW COUNT MINUS 1
0C4C    C551  DPTS 111                        113  FORGET     BR IF G10=1         BR IF DATA CHAIN SET WLR BIT
0C4E    C4D5  DPTS 112                        115  SNSEND     BR IF Z=0           BR IF UCW COUNT 0
0C50    3445  DPTS 113     FORGET                             G0=G0$K40           SET INCORRECT LENGTH
0C52    8D4C  DPTS 114                     DPTR 061  TSTCDA    BR                  BR TO TEST CDA AND SLI FLAGS
0C54    8D56  DPTS 115     SNSEND         DPTR 066  ENDOK     BR                  BR TO CHANNEL END
              DPTS 117    *                 **********************************************************************
              DPTS 118    *                    NO-OP COMMAND
              DPTS 119    *                 **********************************************************************
0C56    DEDE  DPTS 121     NOOPTR             132  ISLCHK     BR IF PRS BIT1=0    BRANCH IF PRINT READY IS OFF
0C58    2EC3  DPTS 125     NOOP                               H0=0$KCC            SET CHANNEL END DEVICE END
0C5A    8D6E  DPTS 126                     DPTR 082  IMMEDI    BR                  GO TO IMMEDIATE COMMAND END
              DPTS 128    *                 **********************************************************************
              DPTS 129    *                       1403 INITIAL SELECTION CHECK
              DPTS 130    *                 **********************************************************************
0C5C    3F85  DPTS 131     CMDREJ                             H1=H1$K80           SET COMMAND REJECT
0C5E    2E23  DPTS 132     ISLCHK                             H0=0$K02            CLEAR UNIT STATUS SET UNIT CHECK
              DPTS 134    *                 **********************************************************************
              DPTS 135    *                    INITIAL SELECTION CHECK OR NON-CHAINED IMMEDIATE
              DPTS 136    *                 **********************************************************************
0C60    1547  DPTS 137     PCENTR                             G1=G1*-K44          RESET COMMAND CHAIN FLAG
0C62    F268  DPTS 138                       141  NOTINT     BR IF D07=0         BR IF LAST COMMAND NOT CHAINED
0C64    3543  DPTS 139                                        G1=G1$K04           TURN ON ACTIVE FLAG IF CHAINED
0C66    8D68  DPTS 140                     DPTR 075  TAKEIN    BR                  GO TO IMMEDIATE CMD INTERRUPT
0C68    5E29  DPTS 141     NOTINT                             V0=H0               SAVE UNIT STATUS
0C6A    B404  DPTS 142                     DCOM 080  CHANOK    BAL                 STORE CH STAT,FLAGS,UNIT STAT
0C6C    52E9  DPTS 143                                        H0=V0               RESTORE UNIT STATUS
0C6E    1485  DPTS 144                                        G0=GC*-K80          RESET SECONDARY BIT
0C70    C174  DPTS 146     TSTPCI             148  NOPCIF     BR IF G14=0         BR IF NO PCI FLAG
0C72    3485  DPTS 147                                        G0=GC$K80           SET PCI INTRPT IN CHAN STATUS
0C74    54F9  DPTS 148     NOPCIF                             H1=G0               MOVE CHANNEL STATUS
0C76    8178  DPTS 149                     CCOM 066  LRSTRB    BAL                 RESTORE P AND I REGS
0C78    A030  DPTS 150                     CCOM 135  STATUS    BR                  BR TO STORE 3'RD HW OF CSW
              DPTS 151    AEND
```

****
* CROSS REFERENCE FOR CSECT DPTS *
****

```
DPTS 043     DCLB 245
DPTS 048     DPTS 045
DPTS 053     DPTS 051
DPTS 062     DPTS 060
DPTS 079     DPTS 071
DPTS 087     DPTS 073
DPTS 092     DPTS 085
DPTS 098     DPTS 089
DPTS 104     DPTS 054  DYPE 177
DPTS 108     DPTS 106
DPTS 109     DPTS 104
DPTS 113     DPTS 111
DPTS 115     DPTS 112
DPTS 121     DPTS 056
DPTS 125     DMCS 019  DMCS 025  ERDR 087
DPTS 131     DMCS 022  DPTS 065  DPTS 066  DPTS 080  DPTS 092
DPTS 132     DPTS 064  DPTS 121
DPTS 137     DPTR 090  ERDR 145
DPTS 141     DPTS 138
DPTS 146     DYPE 153
DPTS 148     DPTS 146
```

DPTT DESCRIPTIVE TEXT

ENTRY POINTS

TEST00 - ENTRY IS MADE AT THIS LABEL FROM THE -DCLA- ROUTINE FOR A START I/O OR TEST I/O TO TEST THE ACTIVE AND SECONDARY BITS TO DETERMINE FURTHER OPERATION.

UCENT - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE-, -EERQ- AND -ERRQ- ROUTINES TO SET UNIT CHECK AND TO STORE A CSW WITH ZERO CHANNEL STATUS WHEN A NOT READY CONDITION IS DETECTED.

RESTES - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE-, -EERQ- AND -ERRQ- ROUTINES TO CLEAR CHANNEL END OR CHANNEL END-DEVICE END.

SHARE - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE-, -EERQ- AND -ERRQ- ROUTINES. DEVICE END IS TESTED.
DE ON - IF INTERRUPT HAS BEEN INITIATED, RESET INTERRUPT LATCH AND STORE STATUS IN THE CSW.
DE OFF - STORE STATUS WITH BUSY SET INTO THE CSW.

DEVE - ENTRY IS MADE AT THIS LABEL FROM THE -DYPE- ROUTINE BECAUSE THE SECONDARY BIT IS ON DURING INITIAL SELECTION FOR START I/O OR TEST I/O. RESET INTERRUPT IF ON, AND STORE STATUS IN THE CSW.

SENCHK - ENTRY IS MADE AT THIS LABEL FROM THE -DPTQ- ROUTINE TO CHECK DEVICE END SENSE DATA. RETURN IS MADE TO -DPTQ-.

RESINB - ENTRY IS MADE AT THIS LABEL FROM THE -DCHN- ROUTINE TO RESET THE INTERRUPT LATCH AND TO CLEAR THE INTERRUPT IN I.B. AND SECONDARY BITS FOR CSW STORE. RETURN IS MADE TO -DCHN-.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                DPTT 001    T                                1403 TEST I/O---START I/O WITH SEC.ON
                DPTT 003    *              ***************************************************************
                DPTT 004    *               TEST I/O ACTIVE=0 SECONDARY=0
                DPTT 005    *              ***************************************************************
0CC8    2482    DPTT 006     TEST00                          SET MODE K=18          SET 1403 MODE
0CCA    8F7A    DPTT 007                   ERRQ 082  RDCHST    BAL    ** GO GET CHAN. AND UNIT STATUS IN G,H
0CCC    D15B    DPTT 008                        020  TEST11    BR IF G15=1            BR IF ACTIVE TEST I00
0CCE    C421    DPTT 009                        046  STTE01    BR IF G00=1             BR ON SEC TO START OR TEST NO A
0CD0    DED4    DPTT 010                        013  UCENT     BR IF PRS1=0           BR IF NOT READY
0CD2    810C    DPTT 011                   CCOM 155  CC0B      BR  **  GO TO COMM MODE RST,CC 0 SET
0CD4    2E23    DPTT 013     UCENT                              H0=0$K02               SET UNIT CHECK IN STA
0CD6    2F07    DPTT 014                                        H1=0                   ZERO OUT CHANNEL STATUS
0CD8    A01E    DPTT 015                   CCOM 123  STATOS    BR ** STORE 0'S IN CSW WITH NEW STATUS
                DPTT 016    *              ***************************************************************
                DPTT 017    *              TEST I/O ACTIVE=1  SECONDARY=1
                DPTT 018    *              ***************************************************************
0CDA    0D08    DPTT 020     TEST11                          RST PRA K=40           RST PRINT REQUEST
0CDC    5FB9    DPTT 021                                     T1=H1                  SAVE SENSE BYTE
0CDE    E401    DPTT 023     RESTES             026  PROGF  0  BR IF G02=1            BR IF PROGRAM ERROR
0CE0    AA19    DPTT 024                        026  PROGF  N  BR IF H0=NZ            BR IF NO STATUS
0CE2    AECA    DPTT 025                   CCOM 162  CC2B      BR                     GO SET BUSY
0C80    154B    DPTT 026     PROGF  0                        G1=G1<K04              RESET ACTIVE BIT
0C82    B448    DPTT 027                        088  RSTINT    BAL                    GO RST S7,INT BUFF BIT,SEC.
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
0C84  DA0F    DPTT 028                033   ZSECON     BR IF H05=1            BR IF DE
0C86  CA0E    DPTT 029                033   ZSECON     BR IF H04=0            BR IF NO CHEND OR NO DEV.END
0C88  E40F    DPTT 030                033   ZSECON     BR IF G02=1            BR IF PRGRAM CHK-DONT SET SEC
0C8A  2085    DPTT 031                                   U0=0$K80               STORE SEC FOR CHE ALONE-UCW
0C8C  8C90    DPTT 032                034   UCWSTR     BR
0C8E  2007    DPTT 033   ZSECON                        U0=0                   DE,CHE STAT-CLEAR CH STAT IN UCW
0C90  7068    DPTT 034   UCWSTR                       STH U AS,D+2           UPDATE UCW  CH STAT AND FLAGS
0C92  C116    DPTT 035                037   DONTCH     BR IF G14=0            BR IF NO PCI FLAG
0C94  3485    DPTT 036                                   G0=G0$K80              SET PCI IN CHAN STAT
0C96  54F9    DPTT 037   DONTCH                          H1=G0                  MOVE CHANNEL STATUS
0C98  7A68    DPTT 038                                   STH T AS,D+2           D=00CC
0C9A  5068    DPTT 039                                   RDH U AS,D+2           UCW COUNT=U
0C9C  526C    DPTT 040                                   RDH V AS,D+0           CCW ADDRESS=V
0C9E  A00E    DPTT 041   CALLH    CCOM 110  FULCSW     BR ** STORE CSW RST MODE, SET CC1
              DPTT 042   *          **********************************************************************
              DPTT 043   *           TEST OR START I/O ACTIVE=0 SECONDARY=1
              DPTT 044   *          **********************************************************************
0CA0  5E29    DPTT 046   STTE01                         V0=H0                  MOVE UNIT STATUS
0CA2  CEB1    DPTT 047                056   EXTDE      BR IF PRS0=1           BR IF EXT DE
0CA4  DA39    DPTT 049   SHARE        061   DEVE       BR IF H0 BIT5=1        BR IF DE.
0CA6  2E15    DPTT 050                                   H0=0$K10               SET BUSY
0CA8  54F9    DPTT 051                                   H1=G0                       MOVE CHANNEL STATUS
0CAA  1F85    DPTT 052                                   H1=H1*-K80             RST SECONDARY
0CAC  F246    DPTT 053                068   DELIC      BR IF D07=0            BR IF START-STORE STATUS ONLY
0CAE  A01E    DPTT 054   GOSTOR   CCOM 123  STATOS     BR                     THIS IS TEST STORE STAT AND 0,S
0CB0  8F00    DPTT 056   EXTDE        071   SENCHK     BAL                    GO CHECK SENSE
0CB2  5F39    DPTT 057                                   V1=H1                  PUT UPDATED SENSE IN V1
0CB4  6664    DPTT 058                                   D=D+2                  UPDATE UCW ADD TO STATUS-CA
0CB6  726A    DPTT 059                                   STH V AS,D-2           STORE UNIT STAT
0CB8  B448    DPTT 061   DEVE         088   RSTINT     BAL  ** GO RST S7,INT BUFF. BIT AND SEC.BIT
0CBA  7460    DPTT 062                                   STH G AS,D+0           UCW CH STATUS
0CBC  54F9    DPTT 063                                   H1=G0                  MOVE CH ST&TUS
0CBE  E243    DPTT 064                066   NOTPTR     BR IF D06=1            BR IF NOT PRINTER
0CC0  0D08    DPTT 065                                   RST PRA K=40           RST PRINTER REQUEST
0CC2  F22F    DPTT 066   NOTPTR       054   GOSTOR     BR IF D07=1            BR IF TEST I/O
0CC4  3E15    DPTT 067                                   H0=H0$K10              SET BUSY
0CC6  A030    DPTT 068   DELIC    CCOM 135  STATUS     BR                     STORE STATUS
              DPTT 069   *          ****************************************
0F00  FE84    DPTT 071   SENCHK       073   CHAN9      BR IF PRS3=0           BR IF NO CHANNEL 12
0F02  3E13    DPTT 072                                   H0=H0$K01              SET UNIT EXCEPTION
0F04  EE8A    DPTT 073   CHAN9        076   DATACH     BR IF PRS2=0           BR IF NO CHANNEL 9
0F06  3E23    DPTT 074                                   H0=H0$K02              SET UNIT CHECK
0F08  3F13    DPTT 075                                   H1=H1$K01              SET CHANNEL 9 IN SENSE
0F0A  CB0E    DPTT 076   DATACH       078   HAMMER     BR IF H14=0            BR IF NO DATA CHECK
0F0C  3E23    DPTT 077                                   H0=H0$K02              SET UNIT CHECK
0F0E  5E3F    DPTT 078   HAMMER                         V1=PRS                 MOVE SENSE
0F10  1393    DPTT 079                                   V1=V1*-K09             LOOK FOR PARITY OR HAMMER CHECK
0F12  F099    DPTT 080                083   SETDE      BR IF LZ=0             BR IF NO ERROR
0F14  3E23    DPTT 081                                   H0=H0$K02              SET UNIT CHECK
0F16  3F15    DPTT 082                                   H1=H1$K10              SET EQUIPMENT CHK IN SENSE
0F18  3E43    DPTT 083   SETDE                          H0=H0$K04              SET DEVICE END
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

0F1A     0D0A     DPTT 084                                    RST PRA K=50           RST BUSY AND REQUEST
0F1C     128E     DPTT 085                                    RTN                    GO BACK TO TEST I/O OR REQ
                  DPTT 087   *                                 ****************************************
3448     F04E     DPTT 088    RSTINT          091   DEVEND    BR IF G07=0            BR IF NO INT IN BUFF
344A     1413     DPTT 089    RESINB                          G0=G0*-K01             RESET INT INDICATOR
344C     0010     DPTT 090                                    RST S7                     RESET INT LATCH
344E     4046     DPTT 091    DEVEND                          U=G                    MOVE CHAN STAT+FLAGS
3450     1485     DPTT 092                                    G0=G0*-K80             RST SEC FOR CSW STORE
3452     128E     DPTT 093                                    RTN
                  DPTT 094   *                                 ********************************
                                                          **********************************
                                                          * CROSS REFERENCE FOR CSECT DPTT *
                                                          **********************************

DPTT 006     DCLA 392
DPTT 013     DPTT 010  DYPE 214  ERRQ 015
DPTT 020     DPTT 008
DPTT 023     DYPE 215  ERRQ 053
DPTT 026     DPTT 023  DPTT 024
DPTT 033     DPTT 028  DPTT 029  DPTT 030
DPTT 034     DPTT 032
DPTT 037     DPTT 035
DPTT 046     DPTT 009
DPTT 049     DYPE 218  ERRQ 054
DPTT 054     DPTT 066
DPTT 056     DPTT 047
DPTT 061     DPTT 049  DYPE 219
DPTT 066     DPTT 064
DPTT 068     DPTT 053
DPTT 071     DPTQ 008  DPTT 056
DPTT 073     DPTT 071
DPTT 076     DPTT 073
DPTT 078     DPTT 076
DPTT 083     DPTT 080
DPTT 088     DPTT 027  DPTT 061
DPTT 089     DCHN 075
DPTT 091     DPTT 088
```

DUMP DESCRIPTIVE TEXT

THE MICRO CORE DUMP IS A NON DESTRUCTIVE PROGRAM STORAGE DUMP. THE DUMP ROUTINE DISPLAYS ONLY ON A NATIVE 1403. ENTRY IS A FUNCTION OF ALTER DISPLAY. SELECTIVE SECTIONS OF PROGRAM STORAGE ARE DISPLAYED WITH ALL OF AUX MODULE ZERO PRECEEDING THE PROGRAM STORAGE SELECTION.

THE ALTER DISPLAY ROUTINE (ALDP) RECOGNIZES A 'DD XXYY' ENTRY AS A DUMP REQUEST. 'XX' DESIGNATES THE START PROGRAM STORAGE MODULE ADDRESS. 'YY' DESIGNATES THE STOP PROGRAM STORAGE MODULE ADDRESS.

THE LINE FORMAT OF THE DUMP ROUTINE IS AS FOLLOWS.

XXXX YYYY YYYY ETC.

THE FIRST TWO BYTES ARE THE ADDRESS. MODULE ZERO IS DENOTED BY THE ABSENCE OF THE FIRST ADDRESS BYTE(UNDERSTOOD AS 00). THERE ARE TWO BLANKS AFTER THE ADDRESS. SIXTEEN ,TWO BYTE GROUPS OF DATA SEPERATED BY BLANKS FOLLOW.

********** EXAMPLE **********

IT IS DESIRED TO SEE PROGRAM STORAGE LOCATIONS 0505 THROUGH, AND INCLUDING, 3050.

PROCEEDURE.

STEP 1. DEPRESS THE ALTER DISPLAY KEY ON THE 2025 CONSOLE
STEP 2. ENTER 'DD0530' ON THE 1052
STEP 3. IF THE PRINTER IS NOT READY, MAKE IT READY.

NOTE. CARRIAGE CONTROL IS PROVIDED BY THE DUMP ROUTINE ASSUMING A STANDARD CARRIAGE TAPE IS ON THE PRINTER (CHANNEL 12 TO INDICATE THE END OF A PAGE AND CHANNEL 1 TO INDICATE THE BEGINNING). A SPECIAL CARRIAGE TAPE MAY HAVE TO BE INSTALLED BEFORE USING THIS ROUTINE.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              DUMP 001    T                      MICRO CORE DUMP ROUTINE
33B2    1F2A  DUMP 002     START                  RST TA K=D2        RST 1052
33B4    9812  DUMP 003                 CCOM 056  LSAVE      BAL                SAVE I REG
33B6    2482  DUMP 004                           SET MODE K=18      SET PRINTER MODE
33B8    3DC2  DUMP 005                           SET PRA K=90       SET PRINT GATE AND BUSY
33BA    2FD3  DUMP 006                           H1=0$K0D             SET UP FOR A SKIP
33BC    0111  DUMP 007                           Z=U1+K01
33BE    F4C3  DUMP 008                 010       AROUND     BR IF AC=1         TEST FOR FF  STOP ADDR
33C0    211B  DUMP 009                           U1=U1+K01
33C2    4252  DUMP 010     AROUND                RDH V DC,9A        ZERO OUT V REG
33C4    4426  DUMP 011                           G=V                ZERO OUT G REG
33C6    A636  DUMP 012                 078       SKIP       BR                 GO DO A SKIP IMMED
31BE    2A75  DUMP 014     INITIL                T0=0$K70           SET UP XLATE TABLE ADDRESS
31C0    2E13  DUMP 019                           H0=0$K01           SET UP PLBAR COUNTER
31C2    25B3  DUMP 020                           G1=0$K0B           SET UP END OF LINE COUNTER
31C4    252D  DUMP 021                           G1=G1+K20
31C6    2F85  DUMP 022                           H1=0$K80           SET UP CONSTANT OF 131 DECIMAL
31C8    2F3B  DUMP 023                           H1=H1+KC3            IN H1 REG FOR PLBAR INCREM
31CA    C053  DUMP 024                 028       GETADR     BR IF GO BIT4=1    BR IF NOT MOD 0 ADDR
31CC    2B45  DUMP 025                           T1=0$K40           LOAD BLANK IN T1
31CE    A606  DUMP 026                 054       LOADBF     BAL                GO PUT BLANK IN BUFFER
31D0    A606  DUMP 027                 054       LOADBF     BAL                GO PUT BLANK IN BUFFER
31D2    4626  DUMP 028     GETADR                D=V                LOAD ADDR INTO D REG
31D4    C05B  DUMP 029                 035       FOUR       BR IF GO BIT4=1    BR IF PROG ADDR
31D6    B1E2  DUMP 030                 039       TWO        BR                 GO LOAD LAST HALF OF MOD 0 ADDR
31F6    C059  DUMP 031     GOSEE       034       PROGDU     BR IF GO BIT4=1    BR IF NOT MOD 0
31F8    5628  DUMP 032                           RDH D AS,V+2       READ OUT OF AUX
31FA    B1DA  DUMP 033                 035       FOUR       BR                 GO LOAD BUFFER
31D8    5638  DUMP 034     PROGDU                RDH D V+2          READ OUT OF PROG
```

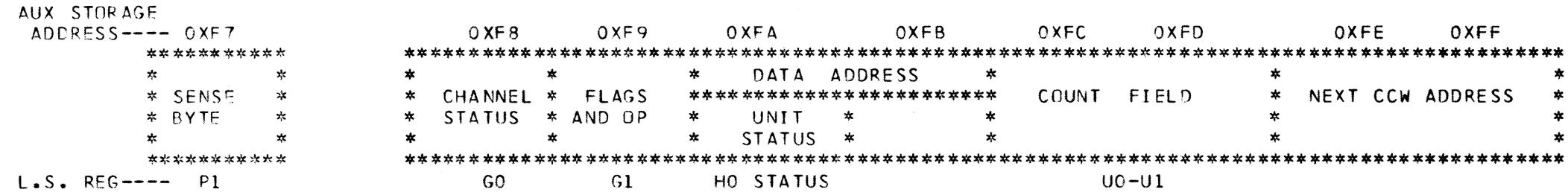
```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
31DA   56B5  DUMP 035      FOUR                          T1=D0XL            SET UP TO UNPACK
31DC   A5FC  DUMP 036                 049      UNPACK     BAL                GO UNPACK CHAR AND LOAD BUFFER
31DE   56B9  DUMP 037                                     T1=D0              SET UP TO UNPACK  CHAR
31E0   A5FC  DUMP 038                 049      UNPACK     BAL                GO UNPACK CHAR AND LOAD BUFFER
31E2   57B5  DUMP 039      TWO                           T1=D1XL            SET UP TO UNPACK  CHAR
31E4   A5FC  DUMP 040                 049      UNPACK     BAL                GO UNPACK CHAR AND LOAD BUFFER
31E6   57B9  DUMP 041                                     T1=D1              SET UP TO UNPACK  CHAR
31E8   A5FC  DUMP 042                 049      UNPACK     BAL                GO UNPACK CHAR AND LOAD BUFFER
31EA   2B45  DUMP 043                                     T1=0$K40           SET UP FOR A BLANK
31EC   A606  DUMP 044                 054      LOADBF     BAL                GO LOAD BUFFER
31EE   C53D  DUMP 045                                     Z=G1<K30           TEST IF DONE WITH ADDR
31F0   C4F4  DUMP 046                 048      NOTADR     BR IF ZNZ
31F2   A606  DUMP 047                 054      LOADBF     BAL                GO LOAD BUFFER
31F4   B1F6  DUMP 048      NOTADR     031      GOSEE      BR
25FC   3BF5  DUMP 049      UNPACK                        T1=T1$KF0          PUMP IN F
25FE   0B61  DUMP 050                                     Z=T1+KC6           TEST FOR 0 THRU 9
2600   F486  DUMP 051                 054      LOADBF     BR IF AC=0         BR IF 0 THRU 9
2602   2B7B  DUMP 052                                     T1=T1+KC7          CONVERT BYTE TO HEX
2604   2BCD  DUMP 053                                     T1=T1+KC0          PUMP IN C
2606   5DA0  DUMP 054      LOADBF                        RDB P1 AS,T+C      READ FROM XLATE TABLE
2608   4FEF  DUMP 055                                     PR=H0              LOAD PLBAR
260A   2D04  DUMP 056                                     SET PRA K=20       SET READ CTRL
260C   2E7B  DUMP 057                                     H0=H0+KC7          INCREMENT PLBAR AND DELAY
260E   4BDF  DUMP 058                                     PR0=P1             LOAD PLB
2610   2EAD  DUMP 059                                     H0=H0+KA0          INCREMENT PLBAR
2612   F497  DUMP 060                 062      OVER88     BR IF AC=1         BR IF OVER 88
2614   6EF3  DUMP 061                                     H0=H0+H1           ADD 131 TO PLBAR
2616   2E2B  DUMP 062      OVER88                        H0=H0+KC2          ADD 2 TO PLBAR
2618   2518  DUMP 063                                     G1=G1+KC1          CHECK FOR END OF DATA
261A   55C9  DUMP 064                                     P0=G1
261C   CC21  DUMP 065                 067      LOADZE     BR IF P0 BIT0=1    BR IF ALL DATA IN BUFFER
261E   128E  DUMP 066                                     RTN
2620   2B45  DUMP 067      LOADZE                        T1=0$K40           SET UP TO LOAD BLANKS
2622   2C2B  DUMP 068                                     P0=P0+KC2          TEST FOR BUFFER FULL
2624   0CBD  DUMP 069                                     Z=P0<KB0
2626   C486  DUMP 070                 054      LOADBF     BR IF ZNZ          BR IF BUFFER NOT FULL
2628   C031  DUMP 071                 075      PRINT      BR IF G0 BIT4=1    BR IF DONE WITH MOD 0
262A   241B  DUMP 072                                     G0=G0+KC1          INCREMENT MOD LINE COUNTER
262C   C030  DUMP 073                 075      PRINT      BR IF G0 BIT4=0    BR IF NOT DONE WITH MOD 0
262E   5029  DUMP 074                                     V0=U0              PUT DUMP START ADDR IN V0
2630   2F93  DUMP 075      PRINT                         H1=C$K09           SET UP FOR PRINT AND SKIP
2632   3D02  DUMP 076                                     SET PRA K=90       SET PRT GATE AND BUSY
2634   FEB8  DUMP 077                 079      CARR       BR IF PRS BIT3=0   BR IF NO CH 12
2636   3F85  DUMP 078      SKIP                          H1=H1$K80          SET UP FOR SKIP
2638   1D00  DUMP 079      CARR                          RST PRA K=80       RST CH 12 AND PRT CK
263A   4DFF  DUMP 080                                     PRC=H1             SET CARR CTRL
263C   FABC  DUMP 081      WAIT       081      WAIT       BR IF PRS BIT7=0   BR IF NO REQ
263E   DB42  DUMP 082                 084      DONT       BR IF H1 BIT5=0    DON'T RESET CH. 12 UNLESS 1ST.
2640   1D00  DUMP 083                                     RST PRA K=80       RESET CH. 12 AND PRT CHK
2642   CD0A  DUMP 084      DONT                          RST PRA K=50       RST PRT REQ AND BUSY
2644   51E9  DUMP 085                                     H0=U1              TEST FOR END OF DUMP
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

2646     6E21     DUMP 086                                    H0=H0<V0
2648     C4D3     DUMP 087                100       QUIT       BR IF Z=0              BR IF DONE
264A     DEBC     DUMP 088                081       WAIT       BR IF PRS BIT1=0       BR IF PRT NOT READY
264C     02CD     DUMP 090                                    Z=V0<KC0               TEST FOR END OF PROGRAM STORAGE
264E     C4D3     DUMP 098                100       QUIT       BR IF Z=0
2650     B1BE     DUMP 099                014       INITIL     BR                     GO PRINT ANOTHER LINE
2652     8174     DUMP 100     QUIT       CCOM 064  LRSTR      BAL                    RESTORE I REG
2654     AB2C     DUMP 101                BSWI 107  RESTRH     BR                     GO TO SOFT STOP
                                                    **********************************
                                                    * CROSS REFERENCE FOR CSECT DUMP *
                                                    **********************************

DUMP 002     ALDP 165
DUMP 010     DUMP 008
DUMP 014     DUMP 099
DUMP 028     DUMP 024
DUMP 031     DUMP 048
DUMP 034     DUMP 031
DUMP 035     DUMP 029  DUMP 033
DUMP 039     DUMP 030
DUMP 048     DUMP 046
DUMP 049     DUMP 036  DUMP 038  DUMP 040  DUMP 042
DUMP 054     DUMP 026  DUMP 027  DUMP 044  DUMP 047  DUMP 051  DUMP 070
DUMP 062     DUMP 060
DUMP 067     DUMP 065
DUMP 075     DUMP 071  DUMP 073
DUMP 078     DUMP 012
DUMP 079     DUMP 077
DUMP 081     DUMP 081  DUMP 088
DUMP 084     DUMP 082
DUMP 100     DUMP 087  DUMP 098
```

DYPE DESCRIPTIVE TEXT

ADDITIONAL INFORMATION REFERENCE
2025 PROCESSING UNIT, F.E. THEORY OF OPERATION MANUAL

1052 UCW



SENSE BYTE

0=COMMAND REJECT
1=INTERVENTION REQUIRED
2=DOES NOT APPLY
3=EQUIPMENT CHECK
4=DOES NOT APPLY
5=DOES NOT APPLY
6=DOES NOT APPLY
7=DOES NOT APPLY

CHANNEL STATUS

0=SECONDARY BIT
1=INCORRECT LENGTH
2=PROGRAM CHECK
3=PROTECTION CHECK
4=0 COUNT BIT
5=CHAN CONTROL CHECK
6=INTERFACE CHECK
7=PR-KB INTERRUPT IN INTERRUPT BUFFER

FLAGS AND OP

0=CHAIN DATA (CD) FLAG
1=CHAIN COMMAND (CC) FLAG
2=SUPPRESS LENGTH INDICATION (SLI) FLAG
3=SKIP FLAG
4=PROGRAM CONTROLLED INTERRUPT (PCI) FLAG
5=ACTIVE BIT
6-7=OP -- 01=WRITE
10=READ OR SENSE
11=WRITE WITH ACR

UNIT STATUS

0=ATTENTION
1=DOES NOT APPLY
2=DOES NOT APPLY
3=BUSY
4=CHANNEL-END
5=DEVICE-END
6=UNIT-CHECK
7=UNIT EXCEPTION

DYPE ENTRY POINTS

TWRREQ -- FROM -BSWI- TO HANDLE A SHARE REQUEST FOR ONE OF THE FOLLOWING.

1. ATTENTION
2. READ SHARE REQUEST
3. WRITE SHARE REQUEST
4. INTERRUPT STKD REQ
5. NOT READY TO READY
6. LOGOUT
7. ALTER/DISPLAY

THE FOLLOWING BITS ARE TESTED AS NECESSARY TO DETERMINE THE STATUS OF THE CURRENT OPERATION.

G1 BIT 5 -- ACTIVE BIT
G0 BIT 0 -- SECONDARY BIT
P0 BIT 4 -- ENDING STATUS BIT

NOATEN -- FROM WRAP TRAP ROUTINE. THIS ENTRY IS USED TO TRY FOR AN INTERRUPT.

TWOFF7 -- START I/O ENTRY. CCW FETCH HAS JUST BEEN COMPLETED IN -DCLB- . READY IS TESTED, THEN THE ROUTINE INITIALIZES FOR THE CURRENT COMMAND.

GOCAL -- FROM 2540 ROUTINE TO SHARE WORD THAT MOVES SENSE REG. AFTER THE WORD IS EXECUTED, THE -DYPE- ROUTINE BRANCHES TO -DPTS- TO DO THE SENSE OPERATION FOR EITHER THE 1052 OR 2540.

TWSEC7 -- FROM I/O INSTRUCTIONS ROUTINE -DCLA- FOR THE FOLLOWING CONDITIONS.

START I/O -- ACT BIT=0, SEC BIT=1
TEST I/O -- ACT BIT=1, SEC BIT=1, CC FLAG=0
TEST I/O -- ACT BIT=0

OUTSTANDING STATUS IS STORED IN THE CSW.

HLTTW -- HALT I/O ENTRY FROM I/O INSTRUCTIONS ROUTINE, -DCLA- . THE ACTIVE AND SECONDARY BITS ARE TESTED TO DETERMINE THE NECESSARY ACTION AND CONDITION CODE SETTING.

STTRSL -- TRANSLATE SUBROUTINE ENTRY. TRANSLATES KEYBOARD CODE CHARACTERS TO EBCDIC. USED BY ALTER/DISPLAY ROUTINE, AS WELL AS DURING A READ REQUEST CYCLE.

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

              DYPE 001    T                  1052 REQUEST ROUTINES
              DYPE 002    ASEQ     ALO7=02
0782   FFA3   DYPE 003     TWRREQ          020  ADPACT     BR IF TU3=1        BR IF ALTER-DISPLAY ACTIVE
0784   FAA5   DYPE 004                     021  LOGREQ     BR IF TT7=1        BR IF LOGOUT REQUEST
0786   9816   DYPE 005                CCOM 058  LSAVEB     BAL                GO SAVE P AND I
0788   A566   DYPE 006                     179  GETUCW     BAL                GET UCW AND SET PR-KB MODE
078A   C826   DYPE 007                     022  NOBUFR     BR IF P04=0        BR IF NOT END STATUS
078C   CE92   DYPE 008     AFTER           011  NOATEN     BR IF TT0=0        BR IF ATTENTION LATCH IS OFF
078E   3E85   DYPE 009     ATTEN                          H0=H0$K80          SET ATTEN IN UNIT STATUS REG
0790   0F10   DYPE 010                                    RST TA K=01        RESET ATTEN LATCH
0792   1F28   DYPE 011     NOATEN                         RST TA K=C2        RST READ, WRITE, AND SHARE REQ
0794   3C83   DYPE 012                                    P0=P0$K08          TURN ON END STATUS PRESENT
0796   8FBE   DYPE 013                DCOM 008  INTRPT     BAL                TRY FOR NORMAL INTRPT (NOT PCI)
0798   CE9C   DYPE 014                     016  QATNO      BR IF TT0=0        BR IF ATTENTION LATCH NOT SET
079A   2F04   DYPE 015                                    SET TA K=20        SET MICROFORCE LATCH
079C   8378   DYPE 016     QATNO           191  PUTUCS     BAL                MOVE IN NEW UNIT STATUS REG
              DYPE 017    *                                                     STORE UCW
079E   FEDC   DYPE 018                     052  GOBSWI     BR IF TT3=0        BR IF ALDP REQUEST LATCH OFF
07A0   9ACC   DYPE 019     ADPREQ     ALDP 037  START      BR                 GO TO ALTER-DISPLAY START
07A2   A6B8   DYPE 020     ADPACT     ALDP 275  RTTYP      BR                 ALDP REQUEST, GO TYPE CHAR
07A4   947A   DYPE 021     LOGREQ     BMCK 092  LOGOUT     BR                 GO HANDLE LOGOUT REQUEST
07A6   D13D   DYPE 022     NOBUFR          034  AC1        BR IF G15=1        BR IF ACTIVE BIT ON
07A8   CE2D   DYPE 023                     025  QATYES     BR IF H00=1        BR IF ATTN BIT IN U-STATUS REG
07AA   2E43   DYPE 024                                    H0=0$K04           SET DEVICE END IN U-STATUS REG
07AC   C413   DYPE 025     QATYES          011  NOATEN     BR IF G00=1        BR IF SECONDARY BIT ON
              DYPE 026    *                    ACTIVE 0  SECONDARY 0
07AE   2485   DYPE 027                                    GO=0$K8C           RST CH STATUS REG, SET SEC BIT
07B0   DE93   DYPE 028                     011  NOATEN     BR IF TT1=1        BR IF NOT-READY TO READY
07B2   2E07   DYPE 029                                    H0=0               CLEAR UNIT STATUS REG
07B4   CE8F   DYPE 030                     009  ATTEN      BR IF TT0=1        BR IF ATTENTION LATCH SET
07B6   FEA1   DYPE 031                     019  ADPREQ     BR IF TT3=1        BR IF ALTER-DISPLAY LATCH ON
07B8   3423   DYPE 032                                    GO=G0$K02          SET INTERFACE CHECK
07BA   8792   DYPE 033                     011  NOATEN     BR
07BC   C471   DYPE 034     AC1             062  OSNS       BR IF G00=1        BR IF SECONDARY BIT ON
              DYPE 035    *                    ACTIVE 1  SECONDARY 0
07BE   2E07   DYPE 036                                    H0=0               CLEAR UNIT STATUS REG
07C0   EEC6   DYPE 037                     040  SINTVN     BR IF TT2=0        BR NOT INTERVENTION REQUIRED
07C2   3D45   DYPE 038                                    P1=P1$K40          SET INTVN REQ IN SENSE REG
07C4   B36E   DYPE 039     DOCRLF          072  CRLF       BR
07C6   8FBA   DYPE 040     SINTVN     DCOM 006  PCIBR      BAL                IF PCI, TRY FOR PCI INTERRUPT
07C8   DFF8   DYPE 041                     066  READOT     BR IF TU1=0        BR IF READ OPERATION
              DYPE 042    *                  WRITE OPERATION
07CA   C05F   DYPE 043                     053  WROBIT     BR IF G04=1        BR IF 0-COUNT BIT SET
07CC   5F38   DYPE 044                                    RDB H1 V+1         READ CHAR AT DATA ADDRESS
07CE   4FFF   DYPE 045                                    TE=H1              SEND CHAR TO TE DATA REG
07D0   0F20   DYPE 046                                    RST TA K=02        ISSUE SHARE RESET
07D2   5006   DYPE 047     WCOMOT                         U=U-1              DECREMENT COUNT
07D4   C4DA   DYPE 048                     051  MORE       BR IF ZNZ          BR IF COUNT = 0
07D6   C57B   DYPE 049                     067  CHAIN1     BR IF G10=1        BR IF CD FLAG
07D8   3483   DYPE 050                                    GO=G0$K08          SET 0 COUNT BIT
07DA   837A   DYPE 051     MORE            192  PUTUCW     BAL                STORE UCW EXCEPT UNIT STATUS
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
07DC  AB20    DYPE 052    GOBSWI   BSWI 107  RESTRH    BR                 RETURN FROM REQUEST
07DE  1483    DYPE 053    WROBIT              G0=G0*-K08 RESET 0 COUNT BIT
07E0  E567    DYPE 054    ADIM          057   SLION     BR IF G12=1        BR IF SLI FLAG ON
07E2  3445    DYPE 055                        G0=G0$K40  SET WLR
07E4  15C5    DYPE 056                        G1=G1*-KC0 BREAK CHAIN (RESET CC,CD FLAGS)
07E6  E145    DYPE 057    SLION         039   DOCRLF    BR IF G16=1        BR IF WRITE ACR OR READ OP
07E8  5DD9    DYPE 058                        P1=P1
07EA  C4F1    DYPE 059                062   OSNS      BR IF Z=0          BR IF SENSE REG IS ZERO
07EC  15C5    DYPE 060                        G1=G1*-KC0 RESET CC, CD FLAGS
07EE  3E23    DYPE 061                        H0=H0$K02  SET UNIT CHECK
07F0  D57B    DYPE 062    OSNS          067   CHAIN1    BR IF G11=1        BR IF CC FLAG
07F2  3EC3    DYPE 063                        H0=H0$K0C  SET CHANNEL END, DEVICE END
07F4  3485    DYPE 064    QFLIP               G0=G0$K80  SET SECONDARY BIT
07F6  8792    DYPE 065                011   NOATEN    BR
07F8  A1D2    DYPE 066    READOT      090   READ      BR
07FA  1485    DYPE 067    CHAIN1              G0=G0*-K80 RESET SECONDARY BIT
07FC  8378    DYPE 068                191   PUTUCS    BAL                MOVE IN NEW UNIT STATUS REG
              DYPE 069  *                                                  STORE UCW
07FE  A598    DYPE 070                162   KKKKKK    BR                 GO RESTORE P, I, THEN CHAIN
              DYPE 071  AEND
336E  2F53    DYPE 072    CRLF                H1=0$K05   SET UP
3370  3F15    DYPE 073                        H1=H1$K10    NEW LINE CHARACTER
3372  1F00    DYPE 074                        RST TA K=80 RESET READ LATCH
3374  2F08    DYPE 075                        SET TA K=40 SET WRITE LATCH
3376  4EEE    DYPE 076                        TE=H1      SEND NL CHAR TO TE DATA REG
3378  0F20    DYPE 077                        RST TA K=02 ISSUE SHARE RESET
337A  3485    DYPE 078                        G0=G0$K80  SET SECONDARY BIT
337C  1483    DYPE 079                        G0=G0*-K08 RESET 0-COUNT BIT
337E  5DD9    DYPE 080                        P1=P1
3380  C487    DYPE 081                084   OKDE      BR IF Z=0          BR IF SENSE REG IS ZERO
3382  3E23    DYPE 082                        H0=H0$K02  SET UNIT CHECK
3384  15C5    DYPE 083                        G1=G1*-KC0 RESET CC, CD FLAGS
3386  D50D    DYPE 084    OKDE          087   EXITA     BR IF G11=1        BR IF CC FLAG
3388  3E83    DYPE 085                        H0=H0$K08  SET CHANNEL END
338A  8FBF    DYPE 086              DCOM 008  INTRPT    BAL                TRY FOR NORMAL INTRP (NOT PCI)
338C  8378    DYPE 087    EXITA         191   PUTUCS    BAL                GET NEW UNIT STATUS, STORE UCW
338E  AB20    DYPE 088              BSWI 107  RESTRH    BR                 RETURN FROM REQUEST
              DYPE 089  *             READ OPERATION
21D2  5AFF    DYPE 090    READ                H1=TI      GET KB CHAR FROM TI REG
21D4  1FC5    DYPE 091                        H1=H1*-KC0   RESET CASE BITS 0,1
21D6  DAF1    DYPE 092                103   ALTCOD    BR IF TT5=1        BR IF ALTERNATE CODING
21D8  CADC    DYPE 093                095   ARNSET    BR IF TT4=0        BR IF NOT KEYBOARD CHECK
21DA  2D15    DYPE 094    SETEQP              P1=C$K10   CLEAR SENSE REG, SET EQUIP CHK
21DC  C06D    DYPE 095    ARNSET          112   EXITB     BR IF G04=1        BR IF 0-COUNT BIT SET
21DE  4806    DYPE 096                        I=U        SAVE COUNT
21E0  957A    DYPE 097                242   STTRSL    BAL                XLATE KB CHAR AND PRINT IT
21E2  4086    DYPE 098                        U=I        RESTORE COUNT
21E4  42CF    DYPE 099                        STPC=P0    SET PROTECT KEY
21E6  F56B    DYPE 1C0                102   SKIP      BR IF G13=1        BR IF SKIP FLAG
21E8  7F38    DYPE 101                        STB H1 V+1 STORE CHAR READ AT DATA ADDR
21EA  87D2    DYPE 102    SKIP          047   WCOMOT    BR                 GO DECREMENT COUNT
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

21F0    0F5B    DYPE 103    ALTCOD                          Z=H1<K05
21F2    C4FD    DYPE 104                 109   RQNOF      BR IF Z=0          BR IF EOB
21F4    0FAB    DYPE 105                                  Z=H1<KCA
21F6    C4DA    DYPE 106                 094   SETEQP     BR IF ZNZ          BR IF NOT CANCEL
21F8    2E13    DYPE 107                                  H0=0$K01           SET UNIT EXCEPTION
21FA    1545    DYPE 108                                  G1=G1*-K40         RESET CC FLAG
21FC    C06F    DYPE 109    RQNOF        111   AD1MA      BR IF G04=1        BR IF 0-COUNT BIT IS SET
21FE    87E0    DYPE 110                 054   AD1M       BR
21EE    B36E    DYPE 111    AD1MA        072   CRLF       BR
21EC    87DE    DYPE 112    EXITB        053   WROBIT     BR
                DYPE 113    *          START I/O  ACTIVE 0  SECONDARY 0
259A    9816    DYPE 114    TWOFF7   CCOM 058   LSAVEB     BAL                SAVE P AND I
259C    54F9    DYPE 115    TWOFF8                          H1=G0              MOVE COMMAND CODE
259E    6664    DYPE 116                                    D=D+2              ADJUST UCW ADR REG TO DATA ADR
25A0    8380    DYPE 117                 195   PUTUCA     BAL                STORE DATA ADR AND COUNT IN UCW
25A2    A566    DYPE 118                 179   GETUCW     BAL                GET UCW, SENSE BYTE, SET MODE
25A4    1485    DYPE 119                                    G0=G0*-K80         RESET SECONDARY BIT
25A6    1C83    DYPE 120                                    P0=P0*-K08         RESET ENDING STATUS BIT
25A8    837A    DYPE 121                 192   PUTUCW     BAL                STORE UCW EXCEPT UNIT STATUS
25AA    2E07    DYPE 122                                    H0=0               CLEAR UNIT STATUS REG
25AC    5FF9    DYPE 123                                    H1=H1
25AE    C4B2    DYPE 124                 126   CONTUE     BR IF ZNZ          BR IF NOT CHAIN DATA
25B0    AB20    DYPE 125             BSWI 107   RESTRH     BR                 GO RESTORE CPU MODE
25B2    1D45    DYPE 126    CONTUE                          P1=P1*-K40         RESET INTERVENTION REQUIRED BIT
25B4    1F28    DYPE 127                                    RST TA K=C2        RST READ, WRITE, AND SHARE REQ
25B6    EEBA    DYPE 128                 130   NONRDY     BR IF TT2=0        BR IF READY
25B8    3D45    DYPE 129                                    P1=P1$K40          SET INTERVENTION REQUIRED
25BA    0F4B    DYPE 130    NONRDY                          Z=H1<K04
25BC    C4DD    DYPE 131                 174   SENSE      BR IF Z=0          BR IF SENSE COMMAND
25BE    EED5    DYPE 132                 147   NOTRDY     BR IF TT2=1        BR IF NOT READY
25C0    0F3B    DYPE 133                                    Z=H1<K03
25C2    C495    DYPE 134                 159   NOOPRA     BR IF Z=0          BR IF NO-OP COMMAND
25C4    2D07    DYPE 135                                    P1=0               CLEAR SENSE REG
25C6    0F1B    DYPE 136                                    Z=H1<K01
25C8    C491    DYPE 137                 164   WRIT       BR IF Z=0          BR IF WRITE COMMAND
25CA    0F9B    DYPE 138                                    Z=H1<K09
25CC    C48F    DYPE 139                 163   WRTACR     BR IF Z=0          BR IF WRITE ACR COMMAND
25CE    0FAB    DYPE 140                                    Z=H1<K0A
25D0    C487    DYPE 141                 170   READOP     BR IF Z=0          BR IF READ COMMAND
25D2    3D85    DYPE 146                                    P1=P1$K80          SET COMMAND REJECT IN SENSE REG
25D4    2E23    DYPE 147    NOTRDY                          H0=0$K02           SET UNIT CHECK IN U-STATUS REG
25D6    15C5    DYPE 148                                    G1=G1*-KC0         RESET CC, CD IN FLAGS/OP REG
25D8    F200    DYPE 149    NOTCCB       151   NOTCCA     BR IF D07=0        BR NOT CHAINED FROM PRIOR CCW
25DA    87F4    DYPE 150                 064   QFLIP      BR                 GO SET SEC BIT, TRY FOR INTRPT
2580    1547    DYPE 151    NOTCCA                          G1=G1*-K44         RESET CC FLAG AND ACTIVE BIT
2582    8378    DYPE 152                 191   PUTUCS     BAL                GET NEW UNIT STATUS, STORE UCW
2584    8C70    DYPE 153           DPTS 146   TSTPCI     BR                 TEST PCI AND STORE CSW STATUS
2594    2EC3    DYPE 159    NOOPRA                          H0=0$K0C           SET CHANNEL END , DEVICE END
2596    D558    DYPE 161                 149   NOTCCB     BR IF G11=0        BR IF NO CC FLAG
2598    8D62    DYPE 162    KKKKKK   DPTR 072   CHAINO     BR                 RESTORE P,I,STORE UCW,CHAIN
258E    3523    DYPE 163    WRTACR                          G1=G1$K02          SET BIT 6 FLAGS/OP REG
```

```
ADDR  WORD  SEQUENCE NO.  LABEL   NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
2590  2F08  DYPE 164      WRIT                        SET TA K=40          SET WRITE LATCH
2592  87DA  DYPE 165                 051   MORE       BR                   GO STORE UCW, RESTORE CPU MODE
2586  3F20  DYPE 170      READOP                      SET TA K=82          SET READ, INITIAL PRTR LATCHES
2588  EB8C  DYPE 171                 173   INNN       BR IF TU6=0          BR IF INITIAL PRTR NOT SET
258A  4FFF  DYPE 172                                   TE=H1              SEND CHARACTER TO 1052 DATA REG
258C  87DA  DYPE 173      INNN        051   MORE       BR                   GO STORE UCW, RESTORE CPU MODE
25DC  3C83  DYPE 174      SENSE                       P0=P0$K08            SET ENDING STATUS BIT
25DE  837A  DYPE 175                 192   PUTUCW     BAL                  STORE UCW EXCEPT UNIT STATUS
25E0  5DF9  DYPE 176      GOCAL                       H1=P1                MOVE SENSE REG
25E2  8C3E  DYPE 177                DPTS 104  SENSE    BR                   GO SHARE PRINTER SENSE ROUTINE
            DYPE 178      *           UCW   FETCH
2566  2791  DYPE 179      GETUCW                      D1=0-K09             SET AUX STORAGE
2568  566D  DYPE 180                                   D0=D0L                 ADDRESS
256A  16E3  DYPE 181                                   D0=D0*-K0E               0XF6
256C  2404  DYPE 182                                   SET MODE K=20        SET PR-KB MODE
256E  5C68  DYPE 183                                   RDH P AS,D+2         GET PROTECT KEY, SENSE BYTE
2570  5468  DYPE 184                                   RDH G AS,D+2         READ CHNL STATUS, FLAGS/OP
2572  5268  DYPE 185                                   RDH V AS,D+2               DATA ADDRESS/UNIT STATUS
2574  5068  DYPE 186                                   RDH U AS,D+2               COUNT FIELD
2576  52E9  DYPE 187                                   H0=V0                MOVE UNIT STATUS
2578  2771  DYPE 188                                   D1=0-K07             RESTORE UCW ADDRESS TO F8
257A  128E  DYPE 189                                   RTN
            DYPE 190      *                                              UCW STORE
0378  5E29  DYPE 191      PUTUCS                      V0=H0                  MOVE UNIT STATUS
037A  2791  DYPE 192      PUTUCW                      D1=0-K09               SET ADDRESS TO 0XF6
037C  7C68  DYPE 193                                   STH P AS,D+2           STORE SENSE BYTE
037E  7468  DYPE 194                                   STH G AS,D+2                 CHANNEL STATUS, FLAGS/OP
0380  7268  DYPE 195      PUTUCA                      STH V AS,D+2                 DATA ADDRESS/UNIT STATUS
0382  7068  DYPE 196                                   STH U AS,D+2                 COUNT FIELD
0384  2771  DYPE 197                                   D1=0-K07             RESTORE UCW ADDRESS TO F8
0386  128E  DYPE 198                                   RTN
            DYPE 199      *                                              * START I/O ACT=0, SEC=1 (D07=0)
            DYPE 200      *                                              * TEST I/O ACT=1, SEC=1, CC FLAG=0
            DYPE 201      *                                              * TEST I/O ACTIVE=0
331C  2791  DYPE 202      TWSEC7                      D1=0-K09               SET ADDRESS TO 0XF6
331E  5B60  DYPE 203                                   RDB T1 AS,D            READ SENSE BYTE
3320  58BB  DYPE 204                                   T1=T1H               RESET SENSE BITS 4-7
3322  7B60  DYPE 205                                   STB T1 AS,D            STORE SENSE BYTE
3324  6664  DYPE 206                                   D=D+2                  INCREMENT UCW ADDRESS TO F8
3326  2404  DYPE 207                                   SET MODE K=20          SET PR-KB MODE
3328  8F7A  DYPE 208                ERRQ 082  RDCHST   BAL                    GO READ CHANNEL AND UNIT STATUS
332A  0F04  DYPE 209                                   RST TA K=20            RESET MICROFORCE LATCH
332C  D139  DYPE 210                     215  ZA1S1    BR IF G15=1            BR IF ACTIVE BIT IS ON
332E  C43B  DYPE 211                     216  ZA0S1    BR IF G00=1            BR IF SECONDARY BIT IS ON
3330  EEB5  DYPE 212                     214  TNRCNT   BR IF TT2=1            BR IF INTERVENTION REQUIRED
3332  810C  DYPE 213                CCOM 155  CCOB     BR                     GO SET CPU MODE, COND CODE 0
3334  8CD4  DYPE 214      TNRCNT   DPTT 013  UCENT    BR                     GO SET UNIT CHECK, STORE CSW
3338  8CDE  DYPE 215      ZA1S1    DPTT 023  RESTES   BR                     GO TEST UNIT STATUS
333A  3623  DYPE 216      ZA0S1                       D0=D0$K02              SET D0 6 FOR PTR RTN TO BR ON
333C  CE37  DYPE 217                     219  ZATEN    BR IF H00=1            BR IF ATTENTION BIT IS ON
333E  8CA4  DYPE 218                DPTT 049  SHARE    BR                     SHARE PTR RTN, THEN STORE CSW
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
3336  8CB8    DYPE 219    ZATEN    DPTT 061 DEVE       BR                 GO RESET I.B., THEN STORE CSW
              DYPE 220    *            HALT IO
246C  8F7A    DYPE 221    HLTTW    ERRQ 082 RDCHST     BAL                GO READ CHANNEL AND UNIT STATUS
246E  2404    DYPE 222                                 SET MODE K=20      SET PR-KB MODE
2470  C463    DYPE 223                  232 ZHACTV      BR IF G00=1        BR IF SECONDARY BIT IS SET
2472  D179    DYPE 224                  228 ZHTERM      BR IF G15=1        BR IF ACTIVE BIT IS SET
2474  FEEC    DYPE 225                  227 ZHAO        BR IF TT2=0        BR IF NO INTERVENTION REQUIRED
2476  ADF6    DYPE 226             CCOM 166 CC3B        BR                 GO SET CPU MODE, COND CODE 3
2460  8A9E    DYPE 227    ZHAO     DCLA 285 STORO       BR                 GO ZERO STATUS, STORE CSW
2478  1F00    DYPE 228    ZHTERM                       RST TA K=80        RESET READ LATCH
247A  2F08    DYPE 229                                 SET TA K=40        SET WRITE LATCH
247C  3483    DYPE 230                                 G0=G0$KC8          SET 0-COUNT BIT
247E  8AC2    DYPE 231             DCLA 279 HOHO       BR                 GO RST CC, CD FLAGS, STORE CSW
2462  D56A    DYPE 232    ZHACTV       236 ZHERE      BR IF G11=0        BR IF NO CC FLAG
2464  3E83    DYPE 233                                 H0=H0$K08          SET CHANNEL END
2466  6664    DYPE 234                                 D=D+2              SET UCW ADDRESS TO UNIT STATUS
2468  7E6A    DYPE 235                                 STH H AS,D-2       STORE UNIT STATUS IN UCW
246A  8AC0    DYPE 236    ZHERE    DCLA 278 DYPADD     BR                 GO TEST CC FLAG
3478  9816    DYPE 237    PRGCHK   CCOM 058 LSAVEB     BAL                GO SAVE P, I
347A  B406    DYPE 238             DCOM 081 REDSTR     BAL                GO STORE UCW
347C  A566    DYPE 239                  179 GETUCW     BAL                GET UCW, SENSE BYTE, SET MODE
347E  8792    DYPE 240                  011 NOATEN     BR                 GO TO ENDING ROUTINE
              DYPE 241    *             NEW TRANSLATE ROUTINE
157A  5AFF    DYPE 242    STTRSL                       H1=TI              GET KEYBOARD CHARACTER
157C  5F19    DYPE 243                                 U1=H1              MOVE CHAR TO WORK REG
157E  11C5    DYPE 244                                 U1=U1*-KC0         RESET CASE BITS -- 0,1
1580  CF10    DYPE 245                  253 LOWCSE     BR IF H10=0        BR IF LOWER CASE
1582  2055    DYPE 246    BACK                         U0=0$K50           SET MODLE ADDR OF TABLE 50XX
1584  21BD    DYPE 247                                 U1=U1+KB0          ADD XXB8
1586  218B    DYPE 248                                 U1=U1+KC8            TO BIAS THE TABLE ADDRESS
1588  5F00    DYPE 249                                 RDB H1 AS,U        GET EBCDIC CHAR FROM TABLE
158A  4FFF    DYPE 250    LOWEQ                        TE=H1              SEND CHAR TO 1052 DATA REG
158C  0F20    DYPE 251                                 RST TA K=02        SHARE RESET TO ALLOW 1052 CYCLE
158E  128E    DYPE 252                                 RTN                RETURN TO LINK ADDRESS
1590  C483    DYPE 253    LOWCSE       246 BACK        BR IF Z=0          BR IF CHAR IS A BLANK
1592  F098    DYPE 254                  257 LOWAQ      BR IF LZNZ         BR IF NOT DASH, AT, AMPERSAND
1594  11CB    DYPE 255                                 U1=U1<KOC          MODIFY TABLE ADDR
1596  C482    DYPE 256                  246 BACK       BR IF ZNZ          UNCONDITIONAL BRANCH
1598  E0B1    DYPE 257    LOWAQ       270 LOWFQ      BR IF HZ=0         BR IF NUMERIC OR POUND SIGN
159A  CB20    DYPE 258                  261 LOWBQ      BR IF H14=0        BR IF NOT KB 8 BIT
159C  DB03    DYPE 259                  246 BACK       BR IF H15=1        BR IF KB 4 BIT
159E  EB2D    DYPE 260                  267 LOWCQ      BR IF H16=1        BR IF KB 2 BIT
15A0  011F    DYPE 261    LOWBQ                        Z=U1<K11
15A2  C4A9    DYPE 262                  265 LOWDQ      BR IF Z=0          BR IF SLASH
15A4  1FBD    DYPE 263                                 H1=H1<KB0          TRANSLATE ALPHA CHAR TO EBCDIC
15A6  C48A    DYPE 264                  250 LOWEQ      BR IF ZNZ
15A8  1F7D    DYPE 265    LOWDQ                        H1=H1<K70          TRANSLATE CHAR TO EBCDIC SLASH
15AA  C48A    DYPE 266                  250 LOWEQ      BR IF ZNZ
15AC  111B    DYPE 267    LOWCQ                        U1=U1<K01          MODIFY TABLE ADDR FOR COMMA,
              DYPE 268    *                                                  DOLLAR SIGN, OR PERIOD
15AE  C482    DYPE 269                  246 BACK       BR IF ZNZ
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

15B0     CB34     DYPE 270    LOWFQ        272    LOWGQ      BR IF H14=0            BR IF NOT KB 8 BIT
15B2     EB39     DYPE 271                 274    LOWHQ      BR IF H16=1            BR IF KB 2 BIT
15B4     1FFD     DYPE 272    LOWGQ                           H1=H1<KF0              TRANSLATE NUMERIC TO EBCDIC
15B6     C48A     DYPE 273                 250    LOWEQ      BR IF ZNZ              MOD TABLE ADDR FOR 0 OR POUND
15B8     114B     DYPE 274    LOWHQ                           U1=U1<K04
15BA     C482     DYPE 275                 246    BACK       BR IF ZNZ
                  DYPE 277    *              FOLLOWING IS THE 1052 XLATE TABLE
                  DYPE 278    ATABLE  ADDR=50B8
50B8     407E     DYPE 279    C                               XAUX'407E4C5E7A6C7D6E5C4D5D7F0000F07B4A6FE2E3E4E5'
50CE     E6E7     DYPE 280    C                               XAUX'E6E7E8E96B4F7C2500006DD1D2D3D4D5D6D7D8D95B5A'
50E4     6015     DYPE 281    C                               XAUX'601516004EC1C2C3C4C5C6C7C8C94B5F5005'
                  DYPE 282    AEND
```

```
                                    **********************************
                                    * CROSS REFERENCE FOR CSECT DYPE *
                                    **********************************

DYPE 003     BSWI 088
DYPE 009     DYPE 030
DYPE 011     BWRP 162  DYPE 008  DYPE 025  DYPE 028  DYPE 033  DYPE 065  DYPE 240
DYPE 016     DYPE 014
DYPE 019     DYPE 031
DYPE 020     DYPE 003
DYPE 021     DYPE 004
DYPE 022     DYPE 007
DYPE 025     DYPE 023
DYPE 034     DYPE 022
DYPE 039     DYPE 057
DYPE 040     DYPE 037
DYPE 047     DYPE 102
DYPE 051     DYPE 048  DYPE 165  DYPE 173
DYPE 052     DYPE 018
DYPE 053     DYPE 043  DYPE 112
DYPE 054     DYPE 110
DYPE 057     DYPE 054
DYPE 062     DYPE 034  DYPE 059
DYPE 064     DYPE 150
DYPE 066     DYPE 041
DYPE 067     DYPE 049  DYPE 062
DYPE 072     DYPE 039  DYPE 111
DYPE 084     DYPE 081
DYPE 087     DYPE 084
DYPE 090     DYPE 066
DYPE 094     DYPE 106
DYPE 095     DYPE 093
DYPE 102     DYPE 100
DYPE 103     DYPE 092
DYPE 109     DYPE 104
DYPE 111     DYPE 109
DYPE 112     DYPE 095
DYPE 114     DCLB 253
DYPE 126     DYPE 124
DYPE 130     DYPE 128
```

****
* CROSS REFERENCE FOR CSECT DYPE *
****

DYPE 147 DYPE 132
DYPE 149 DYPE 161
DYPE 151 DYPE 149
DYPE 159 DYPE 134
DYPE 162 DYPE 070
DYPE 163 DYPE 139
DYPE 164 DYPE 137
DYPE 170 DYPE 141
DYPE 173 DYPE 171
DYPE 174 DYPE 131
DYPE 176 ERDR 104
DYPE 179 DYPE 006 DYPE 118 DYPE 239
DYPE 191 DYPE 016 DYPE 068 DYPE 087 DYPE 152
DYPE 192 DYPE 051 DYPE 121 DYPE 175
DYPE 195 DYPE 117
DYPE 202 DCLA 400
DYPE 214 DYPE 212
DYPE 215 DYPE 210
DYPE 216 DYPE 211
DYPE 219 DYPE 217
DYPE 221 DCLA 277
DYPE 227 DYPE 225
DYPE 228 DYPE 224
DYPE 232 DYPE 223
DYPE 236 DYPE 232
DYPE 237 DCLR 048
DYPE 242 ALDP 304 DYPE 097
DYPE 246 DYPE 253 DYPE 256 DYPE 259 DYPE 269 DYPE 275
DYPE 250 DYPE 264 DYPE 266 DYPE 273
DYPE 253 DYPE 245
DYPE 257 DYPE 254
DYPE 261 DYPE 258
DYPE 265 DYPE 262
DYPE 267 DYPE 260
DYPE 270 DYPE 257
DYPE 272 DYPE 270
DYPE 274 DYPE 271

ECOL DESCRIPTIVE TEXT

ENTRY POINTS

MASK — ENTRY IS MADE TO THIS LABEL FROM THE EPCH ROUTINE WHEN A COLUMN BINARY OPERATION NEEDS A MASK TO START. THE PUNCH ROW IMAGE BUFFER HAS BEEN FILLED WITH ONES, THE DATA ADDRESS AND COUNT HAVE BEEN STORED, 2540 MODE AND CPU ZONE HAVE BEEN SET.

WLR — ENTRY IS MADE TO THIS LABEL FROM THE EPXF ROUTINE WHEN A WRONG LENGTH RECORD HAS BEEN DETECTED. THE WLR BIT IS SET AND STORED IN THE INDICATOR BYTE IN AUXILIARY STORAGE. THE RESIDUAL COUNT IS STORED AND THE ERRQ ROUTINE IS BRANCHED TO FOR A UCW READOUT.

MOVWLR — ENTRY IS MADE FROM THE EPXF ROUTINE AFTER THE PUNCH DATA HAS BEEN TRANSLATED TO ROW IMAGE. THE RESIDUAL COUNT IS STORED AND THE ERRQ ROUTINE IS BRANCHED TO FOR A UCW READOUT.

GOCHAN — ENTRY IS MADE FROM THE ERCX ROUTINE AFTER THE CCW COUNT IS ZERO FOR A DATA CHAINING OPERATION. A BRANCH TO DCLB IS MADE TO PICK UP THE NEXT CCW.

CNTLRG — ENTRY IS FROM THE EPCH ROUTINE DURING A DATA CHAINING OPERATION TO DETERMINE IF THE LAST CCW RESULTED IN AN 8 COLUMN GROUP BEING SET INTO THE PUNCH IMAGE BUFFER. IF AN 8 COLUMN GROUP WAS NOT COMPLETED BY THE LAST CCW, THE INFORMATION IS READ BACK FROM THE BUFFER, COMPLETED, AND STORED BACK INTO THE BUFFER.

NEWMSK N — ENTRY IS MADE TO THIS BRANCH SET FROM THE EPXF ROUTINE TO FETCH A NEW COLUMN MASK AS INDICATED BY THE CONTROL REGISTER -IO-.

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
                ECOL 001  T                  PUNCH COLUMN BINARY AND DATA CHAINING
                ECOL 002  *
1B4A  2485      ECOL 003   MASK                         G0=0$K80               FIRST MASK
1B4C  2AF7      ECOL 004                                T0=0$KFF               RST ALL REG'S
1B4E  5AB9      ECOL 005                                T1=T0
1B50  4EA6      ECOL 006                                H=T
1B52  5B99      ECOL 007                                I1=T1
1B54  5B59      ECOL 008                                G1=T1
1B56  01E3      ECOL 009   AGAIN                        Z=U1*-K0E              MASK FOR ODD CT
1B58  F0E1      ECOL 010                   014  CTEVEN  BR IF LZ=0             BR IF CCW CT EVEN
1B5A  211B      ECOL 011                                U1=U1+K01              MAKE CT EVEN
1B5C  D861      ECOL 012                   014  CTEVEN  BR IF P0 BIT5=1        BR IF TOTAL CT EVEN
1B5E  311B      ECOL 013                                U1=U1-K01 **TOTAL CT ODD, TAKE 1 COL LESS
1B60  E824      ECOL 014   CTEVEN          022  RDDATA  BR IF P0 BIT6=0        BR IF LOWER HALF
1B62  261B      ECOL 015                                D0=D0+K01              WLR CTR + 1
1B64  F4A4      ECOL 016                   022  RDDATA  BR IF AC=0             BR IF NOT WLR
1B66  2DA1      ECOL 017   WLR                          P1=0-K0A               WLR, P=1OF5
1B68  5EC0      ECOL 018                                RDH H AS,P+0           RD IND BYTE
1B6A  3E13      ECOL 019                                H0=H0$K01              WLR INDICATOR
1B6C  7EC0      ECOL 020                                STH H AS,P+0
1B6E  987C      ECOL 021                   059  HALF    BR
1B24  5738      ECOL 022   RDDATA                       RDB D1 V+1             READ DATA
1B26  E72A      ECOL 023                   025  N       BR IF D1 BIT2=0
1B28  6541      ECOL 024                                G1=G1<G0
1B2A  F72E      ECOL 025   N               027  NE      BR IF D1 BIT3=0
1B2C  6B41      ECOL 026                                T1=T1<G0
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
1B2E  C332    ECOL 027    NE          029    NEX        BR IF D1 BIT4=0
1B30  6F41    ECOL 028                                  H1=H1<G0
1B32  D336    ECOL 029    NEX         031    NEXT       BR IF D1 BIT5=0
1B34  6941    ECOL 030                                  I1=I1<G0
1B36  E33A    ECOL 031    NEXT        033    NEXTB      BR IF D1 BIT6=0
1B38  6A41    ECOL 032                                  T0=T0<G0
1B3A  F33E    ECOL 033    NEXTB       035    NEXTBT     BR IF D1 BIT7=0
1B3C  6E41    ECOL 034                                  H0=H0<G0
1B3E  5224    ECOL 035    NEXTBT                        V=V+1                  EVERY OTHER BYTE
1B40  282B    ECOL 036                                  I0=I0+K02              CONTROL REG + 2
1B42  311B    ECOL 037                                  U1=U1-K01              CCW COUNT - 2
1B44  C48D    ECOL 038                41    STORE      BR IF Z=0              BR IF CT=0
1B46  5879    ECOL 039    TEMPRY                        D1=I0
1B48  B309    ECOL 040                127   NEWMSK N   N=D1L                  BREAK FOR NEW MASK
1B0C  75C0    ECOL 041    STORE                         STB G1 AS,P+0          STORE INTO PCH IMAGE
1B0E  2DAB    ECOL 042                                  P1=P1+K0A               BUFFER
1B10  78C0    ECOL 043                                  STB T1 AS,P+0
1B12  2DAB    ECOL 044                                  P1=P1+K0A
1B14  7FC0    ECOL 045                                  STB H1 AS,P+0
1B16  1C1B    ECOL 046                                  P0=PC<K01       **INVERT 1ST 3 OR 2ND 3 IND
1B18  F871    ECOL 047                053   DON2ND     BR IF P0 BIT7=1        BR IF STORE 2ND 3
1B1A  5959    ECOL 048                                  G1=I1                  MOVE 2ND 3 BYTES
1B1C  5AB9    ECOL 049                                  T1=T0
1B1E  5EF9    ECOL 050                                  H1=H0
1B20  2DAB    ECOL 051                                  P1=P1+K0A
1B22  980C    ECOL 052                041   STORE      BR                     GO TO STORE 2ND 3
1B70  2DCD    ECOL 053    DON2ND                        P1=P1+KC0
1B72  2DFB    ECOL 054                                  P1=P1+K0F              POINT TO NEXT 6 COL'S
1B74  5119    ECOL 055                                  U1=U1
1B76  C4FD    ECOL 056                059   HALF       BR IF Z=0              BR IF CCW CT=0
1B78  08AD    ECOL 057                                  Z=I0<KA0               FORCE LOWER TO STOP
1B7A  E0CA    ECOL 058                003   MASK       BR IF HZNZ              IN CASE OF WLR
1B7C  1C2B    ECOL 059    HALF                          P0=P0<K02              INVERT UPPER OR LOWER HALF BIT
1B7E  E814    ECOL 060                095   LOHALF     BR IF P0 BIT6=0        BR IF LOWER HALF
1B80  5629    ECOL 061    MOVWLR                        V0=D0                  MOVE WLR CTR
1B82  71A2    ECOL 062                                  STB U1 DA,AC           STORE RESIDUE COUNT
1B84  5C99    ECOL 063                                  I1=P0                  STORE IX REG
1B86  8F64    ECOL 064              ERRQ 086  RPHUCW     BAL                    READ PUNCH UCW
1B88  8A68    ECOL 065              ERRQ 091  RDIND      BAL                    READ INDICATOR AND SENE
1B8A  2E07    ECOL 066                                  H0=0                   ZERO UNIT STATUS
1B8C  C536    ECOL 067                071   NOCD       BR IF G1 BIT0=0        BR IF NOT CDA
1B8E  F837    ECOL 068                071   NOCD       BR IF P0 BIT7=1        BR IF WLR - OVER COUNT
1B90  52D9    ECOL 069    GOCHAN                        P1=V0                  CDA, STORE WLR CTR IN P1
1B92  95CA    ECOL 070              DCLB 043  CHAIN      BR                     GO CHAIN
1BB6  3485    ECOL 071    NOCD                          G0=G0$K80              SECONDARY BIT
1BB8  5FA0    ECOL 082                                  RDB H1 AS,T+0          READ OLD STACKER SEL. FROM 1OF2
1BBA  7CA8    ECOL 083                                  STH P AS,T+2           STORE NEW STKR
1BBC  9F01    ECOL 085                086   STKSEL N   N=H1 BITS123
1800  2F08    ECOL 086    STKSEL 0                      SET P K=40             POCKET P1(RU031)
1802  980A    ECOL 087                091   COUNT      BR
1804  2F0C    ECOL 088    STKSEL 2                      SET P K=60             POCKET P2(RU31&025)
```

```
ADDR  WORD  SEQUENCE NO.  LABEL      NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
1806  980A  ECOL 089                  091      COUNT      BR
1808  2F0A  ECOL 090      STKSEL 4                        SET P K=50             POCKET RP3(RU31&025)
180A  221B  ECOL 091      COUNT                           V0=V0+K01              INCREMENT WLR CTR
180C  F491  ECOL 092                  094      NOUNCT     BR IF AC=1             BR IF NO UNDER COUNT
180E  3C13  ECOL 093                                      P0=P0$K01              UNDER COUNT,SET WLR INDICATOR
1810  921A  ECOL 094      NOUNCT   EXFR 122      WLRBR      BR                     INTERAGATE CHAINING AND SLI
1B94  5089  ECOL 095      LOHALF                          I0=U0                  RESTORE CONTROL CTR
1B96  51A2  ECOL 096                                      RDB U1 DA,AC           RESTORE CCW ADJUSTED COUNT
1B98  52E2  ECOL 097                                      RDH V DA,BC            RESTORE DATA ADDR
1B9A  D81E  ECOL 098                  100      OKCT       BR IF P0 BIT5=0        BR IF TOTAL CT ODD
1B9C  5224  ECOL 099                                      V=V+1                  TOTAL CT EVEN
1B9E  C1E3  ECOL 100      OKCT                            Z=U1*-K0E              MASK FOR ODD COUNT
1BA0  F0AB  ECOL 101                  106      EVENCT     BR IF LZ=0             BR IF EVEN CT
1BA2  21FF  ECOL 102                                      U1=U1+KFF              MAKE IT EVEN
1BA4  D829  ECOL 103                  105      ODDNOT     BR IF P0 BIT5=1        BR IF TOTAL CT EVEN
1BA6  212B  ECOL 104                                      U1=U1+K02  **TOTAL CT ODD, MAKE 1 COL MORE
1BA8  1C4B  ECOL 105      ODDNOT                          P0=P0<K04  **ODD CT, FLIP TOTAL ODD CT BIT
1BAA  58D5  ECOL 106      EVENCT                          P1=I0XL    **RECOVER PCH IMAGE BUFFER ADDR
1BAC  2DCB  ECOL 107                                      P1=P1+K0C
1BAE  2D3D  ECOL 108                                      P1=P1+K30
1BB0  5889  ECOL 109      CNTLRG                          I0=I0
1BB2  F0BE  ECOL 110                  112      CDMDLE     BR IF LZNZ
1BB4  9B4A  ECOL 111                  003      MASK       BR
1BBE  2D3D  ECOL 112      CDMDLE                          P1=P1+K30  **CD STARTED FROM THE MIDDLE OF
1BC0  2D2B  ECOL 113                                      P1=P1+K02   **ROW FORM BYTES, CHANGE ADDR
1BC2  5FC0  ECOL 114      READ                            RDB H1 AS,P+0              TO READ THEM OUT BACKWARDS
1BC4  3D9B  ECOL 115                                      P1=P1-K09
1BC6  5BC0  ECOL 116                                      RDB T1 AS,P+0
1BC8  3D9B  ECOL 117                                      P1=P1-K09
1BCA  55C0  ECOL 118                                      RDB G1 AS,P+0
1BCC  1C1B  ECOL 119                                      P0=P0<K01       **INVERT 1ST 3 OR 2ND 3 IND
1BCE  F852  ECOL 120                  122      SECOND     BR IF P0 BIT7=0        BR IF RD 1ST 3 BYTES
1BD0  9B46  ECOL 121                  039      TEMPRY     BR                     RD ALL 6 BYTES, GO
1BD2  5FE9  ECOL 122      SECOND                          H0=H1                  MOVE THE LOWER 3 BYTE
1BD4  5BA9  ECOL 123                                      T0=T1
1BD6  5599  ECOL 124                                      I1=G1
1BD8  3D9B  ECOL 125                                      P1=P1-K09              SUBT 0A ONCE MORE
1BDA  9BC2  ECOL 126                  114      READ       BR                **GO READ THE UPPER 3 BYTES
1C00  C822  ECOL 127      NEWMSK 0    144      NOTPOL     BR IF P0 BIT4=0        BR IF FROM EPXF
1C02  980C  ECOL 128                  C41      STORE      BR              **DONE 8 COLUMN,GO STORE THEM
1C04  2445  ECOL 129      NEWMSK 2                        G0=0$K40
1C06  9C1E  ECOL 130                  142      BRNCH      BR
1C08  2425  ECOL 131      NEWMSK 4                        G0=0$K20
1C0A  9C1E  ECOL 132                  142      BRNCH      BR
1C0C  2415  ECOL 133      NEWMSK 6                        G0=0$K10
1C0E  9C1E  ECOL 134                  142      BRNCH      BR
1C10  2483  ECOL 135      NEWMSK 8                        G0=0$K08
1C12  9C1E  ECOL 136                  142      BRNCH      BR
1C14  2443  ECOL 137      NEWMSK A                        G0=0$K04
1C16  9C1E  ECOL 138                  142      BRNCH      BR
1C18  2423  ECOL 139      NEWMSK C                        G0=0$K02
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

1C1A    9C1E    ECOL 140                  142   BRNCH      BR
1C1C    2413    ECOL 141    NEWMSK E              GO=0$KO1
1C1E    C824    ECOL 142    BRNCH         145   POLNOT     BR IF PO BIT4=0        BR IF FROM EPXF
1C20    9B56    ECOL 143                  009   AGAIN      BR
1C22    9C9C    ECOL 144    NOTPOL    EPXF 069  STORE      BR              **GOT NEW MASK BACK TO PXFR
1C24    9CD2    ECOL 145    POLNOT    EPXF 016  UPDATE     BR              **GOT NEW MASK BACK TO PXFR
                                                 ***********************************
                                                 * CROSS REFERENCE FOR CSECT ECOL *
                                                 ***********************************

ECOL 003    ECOL 058  ECOL 111  EPCH 104
ECOL 009    ECOL 143
ECOL 014    ECOL 010  ECOL 012
ECOL 017    EPXF 019
ECOL 022    ECOL 014  ECOL 016
ECOL 025    ECOL 023
ECOL 027    ECOL 025
ECOL 029    ECOL 027
ECOL 031    ECOL 029
ECOL 033    ECOL 031
ECOL 035    ECOL 033
ECOL 039    ECOL 121
ECOL 041    ECOL 038  ECOL 052  ECOL 128
ECOL 053    ECOL 047
ECOL 059    ECOL 021  ECOL 056
ECOL 061    EPXF 082
ECOL 069    ERCX 038
ECOL 071    ECOL 067  ECOL 068
ECOL 086    ECOL 085
ECOL 091    ECOL 087  ECOL 089
ECOL 094    ECOL 092
ECOL 095    ECOL 060
ECOL 100    ECOL 098
ECOL 105    ECOL 103
ECOL 106    ECOL 101
ECOL 109    EPCH 087
ECOL 112    ECOL 110
ECOL 114    ECOL 126
ECOL 122    ECOL 120
ECOL 127    ECOL 040  EPXF 040
ECOL 142    ECOL 130  ECOL 132  ECOL 134  ECOL 136  ECOL 138  ECOL 140
ECOL 144    ECOL 127
ECOL 145    ECOL 142
```

EPCH DESCRIPTIVE TEXT

ENTRY POINTS

NPCH — ENTRY IS MADE HERE FROM THE ERDR ROUTINE DURING A PUNCH WRITE START I/O OR A PFR. THE INDICATOR AND SENSE BYTES ARE SET UP AND SAVED IN LOCAL STORAGE.

PICKUP — ENTRY IS MADE FROM THE ERDR ROUTINE TO TEST FOR UNUSUAL COMMAND SEQUENCE. THE READ INDICATOR AND WLR COUNTER ARE SET, ANY EXCESS BYTE COUNT IS STORED IN AUX 0XBE.

HIGHCT — ENTRY IS MADE FROM THE ERDR ROUTINE WITH A NEW CCW BYTE COUNT, TO DETERMINE IF AN EXCESS COUNT HAS BEEN GIVEN. IF THERE IS AN EXCESS COUNT, THE EXCESS IS STORED IN AUX 0XBE.

NATIVE 2540 PUNCH UCW FORMAT AND AUX LOCATIONS

00E8	00E9	00EA		00EC-ED	00EE-FF
CHANNEL STATUS	CHANNEL FLAGS	UNIT STATUS	NOT USED	RESIDUAL COUNT	NEXT CCW ADDRESS

INDICATORS AND SENSE BYTE AND AUX LOCATIONS

10F4	10F5	10F2
INDICATOR	SENSE BYTE	OLD STACKER

CHANNEL STATUS
0=SECONDARY
1=WLR
2=PROG CHK
3=PROT CHK
4=NOT USED
5=NOT USED
6=INTFCE CHK
7=INT IN BFR

CHANNEL FLAGS
0=CHAIN DATA
1=CHAIN CMD
2=SLI
3=SKIP
4=PCI
5=ACTIVE
6=NOT USED
7=NOT USED

UNIT STATUS
0=NOT USED
1=NOT USED
2=NOT USED
3=BUSY
4=CH END
5=DEV END
6=UNIT CHK
7=UNIT EXCEPT

INDICATOR
0=CHAIN DATA
1=STK SEL
2=STK SEL
3=0
4=PFR READ
5=COUNT STOR
6=PFR WRITE
7=WLR

SENSE BYTE
0=CMD REJ
1=NOT RDY
2=NOT USED
3=EQUIP CHK
4=VALIDITY CHK
5=NOT USED
6=UNUSUAL CMD
7=NOT USED

OLD STACKER
0=NOT USED
1=STK SEL
2=STK SEL
3=0
4=NOT USED
5=NOT USED
6=NOT USED
7=NOT USED

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EPCH 001	T			PUNCH SIO CMD DECODE & CHAINING	
		EPCH 002	*			FOLLOWING INFORMATION--	
		EPCH 003	*				
		EPCH 004	*			U REG CONTAINS - COUNT	
		EPCH 005	*			V REG CONTAINS - DATA ADDRESS	
		EPCH 006	*			G0 REG CONTAINS - CHANNEL STATUS	
		EPCH 007	*			G1 REG CONTAINS - FLAG	
		EPCH 008	*			H0 REG CONTAINS - COMMAND BYTE	
		EPCH 009	*			(H0=00 IF DATA CHAINING)	
14D4	1F00	EPCH 010	NPCH			RST P K=80	RST PCH & VAL CHKS(RU031)
14D6	FA74	EPCH 014		044	READ	BR IF H0 BIT7=0	BR TO CHECK IF READ COMMAND
14D8	6CF7	EPCH 016				P0=P0*H1	CLEAR STKR

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
14DA  EA53    EPCH 017               032  COMREJ    BR IF H0 BIT6=1    BR IF 7,6 COMBINATION
14DC  CE62    EPCH 018               021  STK1BT    BR IF H0 BIT0=0    BR IF STKR P1 OR P2
14DE  DE53    EPCH 019               032  COMREJ    BR IF H0 BIT1=1    BR IF 0&1 BITS IN COMMAND
14E0  3C45    EPCH 020                            P0=P0$K40             RP3
14E2  DE66    EPCH 021  STK1BT       023  SSDONE    BR IF H0 BIT1=0
14E4  3C25    EPCH 022                            P0=P0$K20             P2
14E6  E853    EPCH 023  SSDONE       032  COMREJ    BR IF P0 BIT6=1    BR IF PREVIOUS PFR WRITE
14E8  CA46    EPCH 024               034  NOTPFR    BR IF H0 BIT4=0    BR IF THIS CMD NOT PFR
14EA  3C23    EPCH 026                            P0=P0$K02             PFR WRITE CMD,SET INDICATOR
14EC  2F4C    EPCH 027  ORALSO                    SET P K=04            SET PFR RESTART (RU025)
14EE  2462    EPCH 028                            SET MODE K=16         PUNCH MODE AND ZONE
14F0  1643    EPCH 029                            D0=D0*-K04            PFR INDICATOR FOR PCH TRAP(ETRP)
14F2  94CC    EPCH 030               037  MODE      BR
14D2  99B4    EPCH 032  COMREJ  ERDR 138  CMDREJ    BR
14C6  C86D    EPCH 034  NOTPFR       027  ORALSO    BR IF P0 BIT4=1    BR IF PREVIOUSLY PFR READ
14C8  2462    EPCH 035                            SET MODE K=16         PCH MODE 2500 ZONE
14CA  3643    EPCH 036                            D0=D0$K04             SET NOT PFR INDICATOR
14CC  24C2    EPCH 037  MODE                      SET MODE K=10
14CE  1C83    EPCH 038                            P0=P0*-K08            SET FEED INDICATOR(RST PFR READ)
14D0  950C    EPCH 042               057  SETCTR    BR
14F4  EA52    EPCH 044  READ         032  COMREJ    BR IF H0 BIT6=0
14F6  DA53    EPCH 045               032  COMREJ    BR IF H0 BIT5=1    BR IF 5-6 BIT COMBINATION
14F8  0E49    EPCH 046                            Z=H0+K40
14FA  F4D2    EPCH 047               032  COMREJ    BR IF AC=0         BR IF 0&1 BITS NOT ON
14FC  EE53    EPCH 048               032  COMREJ    BR IF H0 BIT2=1    BR IF COL BIN
14FE  EF84    EPCH 049               053  PICKUP    BR IF PS BIT2=0    BR IF NOT UNIT EXCEPTION
1500  2C07    EPCH 050                            P0=0                  PFR UNIT EXCEPTION,ALLOW FEED
1502  99B0    EPCH 051          ERDR 133  UEXPFR    BR                 SHARE UNIT EXCEPTION READER
1504  C808    EPCH 053  PICKUP       055  SETP4     BR IF P0 BIT4=0    BR IF PREVIOUS CMD IS A FEED
1506  2D23    EPCH 054                            P1=0$K02              PREV A READ,SET UNUSUAL CMD BIT
1508  3C83    EPCH 055  SETP4                     P0=P0$K08             SET READ INDICATOR
150A  1C23    EPCH 056                            P0=P0*-K02            RST FEED OR PFR WRITE INDICATOR
150C  3659    EPCH 057  SETCTR                    D0=0-K50              SET WLR D0=AF
150E  5009    EPCH 058  HIGHCT                    U0=U0
1510  C4A3    EPCH 059               068  HICTO     BR IF Z=0          BR IF NO EXCESS COUNT
1512  3C53    EPCH 060                            P0=P0$K05             EXESS COUNT & WLR INDICATORS
1514  4A06    EPCH 061                            T=U                   MOVE COUNT TO T REG
1516  2007    EPCH 062                            U0=0                  GET
1518  21A5    EPCH 063                            U1=0$KA0               EXCESS COUNT
151A  2002    EPCH 064                            SET S3
151C  7B19    EPCH 065                            T1C=T1-U1+C              AND
151E  7A09    EPCH 066                            T0C=T0-U0+C               STORE
1520  7AF2    EPCH 067                            STH T DA,BE                 IT
1522  B3E8    EPCH 068  HICTO   ERRQ 101  STIND     BAL                STORE  .
1524  71A2    EPCH 069                            STB U1 DA,AC          STORE ADJUSTED COUNT
1526  72E2    EPCH 070                            STH V DA,BC           STORE DATA ADDRESS
1528  F72C    EPCH 071               074  STAY      BR IF D1 BIT3=0    BR IF PUNCH
152A  99C4    EPCH 072  GOBACK  ERDR 146  BACK      BR                 READER,GO TO ERDR ROUTINE
152C  C82B    EPCH 074  STAY         072  GOBACK    BR IF P0 BIT4=1    BR IF PFR READ(CC OR SIO)
152E  CC42    EPCH 075               088  CCSIO     BR IF P0 BIT0=0    BR IF NOT CDA
1530  5CD9    EPCH 079                            P1=P0                 MOVE IND TO P1 TO SAVE CDA IND
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
1532   59C9    EPCH 080                                   P0=I1              CDA,RESTORE P0
1534   C85E    EPCH 081                102        GOEPXF  BR IF P0 BIT4=0    CDA FOR NO COL
1536   5809    EPCH 082                                   U0=I0              STORE CONTROL REG
1538   D83F    EPCH 083                086        TOTEVN  BR IF P0 BIT5=1    BR IF TOTAL COUNT EVEN
153A   5224    EPCH 084                                   V=V+1              LAST TOTAL COUNT
153C   282B    EPCH 085                                   I0=I0+K02            WAS ODD
153E   58D5    EPCH 086  TOTEVN                           P1=I0XL            RECOVER BUFFER ADDRESS
1540   9BB0    EPCH 087                ECOL 109   CNTLRG  BR
1542   2B07    EPCH 088  CCSIO                            T1=0               CC OR SIO, T=1000
1544   2C45    EPCH 089                                   P0=0$K40
1546   2DC3    EPCH 090                                   P1=0$K0C           P=400C
1548   28F7    EPCH 091                                   I0=0$KFF
154A   5899    EPCH 092                                   I1=I0              I=FFFF
154C   54A0    EPCH 093  LOOP                             RDH G AS,T+0       XFER PCH IMAGE AREA(1000) TO
154E   78A8    EPCH 094                                   STH I AS,T+2        PCH CHK AREA (400C) AND STORE
1550   74C8    EPCH 095                                   STH G AS,P+2          FF BACK INTO PCH IMAGE(1000)
1552   CD4C    EPCH 096                093        LOOP    BR IF P1 BIT0=0    BR IF NOT DONE
1554   D94C    EPCH 097                093        LOOP    BR IF P1 BIT5=0    BR IF NOT DONE
1556   5884    EPCH 098                                   I=I+1              DONE,I=0, P=4084
1558   3CE9    EPCH 099                                   P0=0-KE0
155A   2D07    EPCH 100                                   P1=0               P=1F00
155C   EE63    EPCH 101                104        GOECOL  BR IF H0 BIT2=1
155E   1CA3    EPCH 102  GOEPXF                           P0=P0*-KCA         P=15XX, DATA MODE 1 WRITE
1560   9CC0    EPCH 103                EPXF 003   PXFR    BR                 SIO, CC OR CDA
1562   9B4A    EPCH 104  GOECOL        ECOL 003   MASK    BR                 COL BIN,SIO,CC
```

```
                                    **********************************
                                    * CROSS REFERENCE FOR CSECT EPCH *
                                    **********************************
EPCH 010    ERDR 095
EPCH 021    EPCH 018
EPCH 023    EPCH 021
EPCH 027    EPCH 034
EPCH 032    EPCH 017  EPCH 019  EPCH 023  EPCH 044  EPCH 045  EPCH 047  EPCH 048
EPCH 034    EPCH 024
EPCH 037    EPCH 030
EPCH 044    EPCH 014
EPCH 053    EPCH 049  ERDR 130
EPCH 055    EPCH 053
EPCH 057    EPCH 042
EPCH 058    ERDR 059
EPCH 068    EPCH 059
EPCH 072    EPCH 074
EPCH 074    EPCH 071
EPCH 086    EPCH 083
EPCH 088    EPCH 075
EPCH 093    EPCH 096  EPCH 097
EPCH 102    EPCH 081
EPCH 104    EPCH 101
```

EPXF DESCRIPTIVE TEXT

ENTRY POINTS

PXFR — ENTRY IS MADE HERE FROM THE EPCH ROUTINE TO SET UP THE WORK AREA AND COUNTER. THE DATA TO BE PUNCHED IS TRANSFERED AND SET INTO THE ROW IMAGE BUFFER IN COMPLEMENT FORM. SEE FEMDM DIAGRAM 5-114.

UPDATE — ENTRY IS MADE HERE FROM THE ECOL ROUTINE TO INCREMENT THE WLR COUNTER AFTER A NEW COLUMN MASK HAS BEEN SET. IF THE WLR COUNTER OVERFLOWS, THE ECOL ROUTINE IS BRANCHED TO AND THE WLR BIT IS SET.

STORE — ENTRY IS MADE HERE FROM THE ECOL ROUTINE TO STORE THE 12, 11, 0, 8, AND 9 ROW WORK REGISTERS INTO THE ROW IMAGE BUFFER. IF THE CCW COUNT IS NOW ZERO, A BRANCH IS TAKEN TO THE ECOL ROUTINE. IF THE COUNT IS NOT ZERO, A NEW COLUMN MASK IS SET UP AND THE BUILDING OF THE PUNCH ROW IMAGE CONTINUES.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EPXF 001	T			PUNCH TRANSFER(EBCDIC TO ROW FORM) & CHAINING	
		EPXF 002	*				
1CC0	5109	EPXF 003	PXFR			U0=U1	MOVE CCW CT
1CC2	2A65	EPXF 007				T0=0$K60	X'LATE TABLE ADDR FOR NOT 24K
1CC4	CD5B	EPXF 009		024	CDA	BR IF P1 BIT0=1	BR IF DATA CHAIN
1CC6	21F7	EPXF 010	CLEAR			U1=0$KFF	CLEAR
1CC8	5179	EPXF 011				D1=U1	ALL THE
1CCA	51E9	EPXF 012				H1=U1	ACCUMULATING
1CCC	51E9	EPXF 013				H0=U1	REGISTERS
1CCE	5199	EPXF 014				I1=U1	
1CD0	2485	EPXF 015				G0=0$K80	FIRST MASK
1CD2	261B	EPXF 016	UPDATE			D0=D0+K01	INCREMENT WLR CTR
1CD4	F4B8	EPXF 017		020	READ	BR IF AC=0	
1CD6	5C19	EPXF 018				U1=U0	
1CD8	9B66	EPXF 019		ECOL	017 WLR	BR	
1CB8	5B38	EPXF 020	READ			RDB T1 V+1	READ FROM MAIN STORE
1CBA	55A0	EPXF 021				RDB G1 AS,T+0	XLATE TO CONDENSED
1CBC	157B	EPXF 022				G1=G1<K07	INVERTS BITS 4,2,1
1CBE	9103	EPXF 023		041	BIT421 N	N=G1 BITS567	
1CDA	58D5	EPXF 024	CDA			P1=IOXL	**CHAINING, RECOVER BUFFER ADR
1CDC	5889	EPXF 025				I0=IC	
1CDE	F0C7	EPXF 026		010	CLEAR	BR IF LZ=0	**BR IF LAST CHAIN ENDED AT 8 BYTES BOUNDARY
		EPXF 027	*				
1CE0	51C0	EPXF 028				RDB U1 AS,P+0	LAST CHAIN(OR SIO)
1CE2	2DAB	EPXF 029				P1=P1+K0A	ENDED IN THE MIDDLE
1CE4	57C0	EPXF 030				RDB D1 AS,P+0	OF A 8 BYTES BOUN-
1CE6	2DAB	EPXF 031				P1=P1+K0A	DARY, READ THE 12,
1CE8	59C0	EPXF 032				RDB I1 AS,P+0	11, 0, 8, & 9TH
1CEA	2D5D	EPXF 033				P1=P1+K50	BYTES OUT
1CEC	5FC0	EPXF 034				RDB H1 AS,P+0	
1CEE	2DAB	EPXF 035				P1=P1+K0A	
1CF0	55C0	EPXF 036				RDB G1 AS,P+0	
1CF2	55E9	EPXF 037				H0=G1	
1CF4	58D5	EPXF 038				P1=IOXL	RECOVER BUFFER ADDR

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
1CF6    5859    EPXF 039    TEMPRY                         G1=IO
1CF8    B109    EPXF 040                ECOL 127  NEWMSK N N=G1L              GO GET NEW MASK
1900    2DAB    EPXF 041    BIT421 0                       P1=P1+K0A          ADD 60(90) - 7 PUNCH
1902    2DAB    EPXF 042    BIT421 1                       P1=P1+K0A          ADD 50(80) - 6 PUNCH
1904    2DAB    EPXF 043    BIT421 2                       P1=P1+K0A          ADD 46(70) - 5 PUNCH
1906    2DAB    EPXF 044    BIT421 3                       P1=P1+K0A          ADD 3C(60) - 4 PUNCH
1908    2DAB    EPXF 045    BIT421 4                       P1=P1+K0A          ADD 32(50) - 3 PUNCH
190A    2DAB    EPXF 046    BIT421 5                       P1=P1+K0A          ADD 28(40) - 2 PUNCH
190C    9C74    EPXF 047    BIT421 6        049  ADD1E     BR                 GO ADD 1E(30)-1 PUNCH
190E    9C80    EPXF 048    BIT421 7        055  BYTES     BR                 NO PUNCH IN 1 THRU 7
1C74    2D1D    EPXF 049    ADD1E                          P1=P1+K1C          ADD 1E TO BUFFER ADDR
1C76    2DEB    EPXF 050                                   P1=P1+K0E
1C78    5BC0    EPXF 051                                   RDB T1 AS,P+0      STORE
1C7A    6841    EPXF 052                                   T1=T1<G0             1 THRU 7
1C7C    7BC0    EPXF 053                                   STB T1 AS,P+0         PUNCH
1C7E    58D5    EPXF 054                                   P1=IOXL            RECOVER PCH BFR ADDR
1C80    C504    EPXF 055    BYTES           057  N         BR IF G1 BIT0=0
1C82    6141    EPXF 056                                   U1=U1<G0           12TH PUNCH
1C84    D508    EPXF 057    N               059  NE        BR IF G1 BIT1=0
1C86    6741    EPXF 058                                   D1=D1<G0           11TH PUNCH
1C88    E50C    EPXF 059    NE              061  NEX       BR IF G1 BIT2=0
1C8A    6941    EPXF 060                                   I1=I1<G0           0TH PUNCH
1C8C    F510    EPXF 061    NEX             063  NEXT      BR IF G1 BIT3=0
1C8E    6F41    EPXF 062                                   H1=H1<G0           8TH PUNCH
1C90    C114    EPXF 063    NEXT            065  DONE      BR IF G1 BIT4=0
1C92    6E41    EPXF 064                                   H0=H0<G0           9TH PUNCH
1C94    282B    EPXF 065    DONE                           I0=I0+K02          ADD 2 TO CONTROL CTR
1C96    20FF    EPXF 066                                   U0=U0+KFF          DECREMENT CCW CT
1C98    C49D    EPXF 067                    069  STORE     BR IF Z=0
1C9A    9CF6    EPXF 068                    039  TEMPRY    BR                 NOT DONE,GET NEW MASK
1C9C    71C0    EPXF 069    STORE                          STB U1 AS,P+0      STORE 12TH BYTE
1C9E    2DAB    EPXF 070                                   P1=P1+K0A
1CA0    77C0    EPXF 071                                   STB D1 AS,P+0      STORE 11TH BYTE
1CA2    2DAB    EPXF 072                                   P1=P1+K0A
1CA4    79CC    EPXF 073                                   STB I1 AS,P+0      STORE 0TH BYTE
1CA6    2D5D    EPXF 074                                   P1=P1+K50
1CA8    7FC0    EPXF 075                                   STB H1 AS,P+0      STORE 8TH BYTE
1CAA    2DAB    EPXF 076                                   P1=P1+K0A
1CAC    5EF9    EPXF 077                                   H1=H0
1CAE    7FC0    EPXF 078                                   STB H1 AS,P+0      STORE 9TH BYTE
1CB0    58D5    EPXF 079                                   P1=IOXL            RESTORE BFR ADDR
1CB2    5019    EPXF 080                                   U1=U0
1CB4    C4C6    EPXF 081                    010  CLEAR     BR IF ZNZ          BR IF CTR NOT ZERO
1CB6    9B80    EPXF 082                ECOL 061  MOVWLR    BR                 FINISHED
                EPXF 083    *                              NATIVE 2540 PUNCH TRANSLATE TABLE
                EPXF 085    ATABLE  ADDR=6000
6000    B989    EPXF 089    C                              XAUX'B9898A8B8C8D8E8F98999A9B9C9D9E9FD9494A4B4C4D'
6016    4E4F    EPXF 090    C                              XAUX'4E4F58595A5B5C5D5E5F79292A2B2C2D2E2F38393A3B'
602C    3C3D    EPXF 091    C                              XAUX'3C3D3E3FF9090A0B0C0D0E0F18191A1B1C1D1E1F00A9'
6042    AAAB    EPXF 092    C                              XAUX'AAABACADAEAFB8919293949596978 0C9CACBCCCDCECF'
6058    D851    EPXF 093    C                              XAUX'D8515253545556574 0216A6B6C6D6E6F7831C0333435'
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

606E     3637     EPXF 094   C                               XAUX'3637E0E9EAEBECEDEEEFF81112131415161718 1A1A2A3'
6084     A4A5     EPXF 095   C                               XAUX'A4A5A6A7B0A8B2B3B4B5B6B7D1C1C2C3C4C5C6C7D0C8'
609A     D2D3     EPXF 096   C                               XAUX'D2D3D4D5D6D7716162636465666770687273747576 77'
60B0     F1E1     EPXF 097   C                               XAUX'F1E1E2E3E4E5E6E7F0E8F2F3F4F5F6F7A0818283848 5'
60C6     8687     EPXF 098   C                               XAUX'86879088BABBBCBDBEBF60414243444546475048DADB'
60DC     DCDD     EPXF 099   C                               XAUX'DCDDDEDF326922232425262730287A7B7C7D7E7F2001'
60F2     0203     EPXF 100   C                               XAUX'02030405060710 08FAFBFCFDFEFF'
                  EPXF 101   AEND

                                                  ************************************
                                                  * CROSS REFERENCE FOR CSECT EPXF   *
                                                  ************************************

EPXF 003    EPCH 103
EPXF 010    EPXF 026  EPXF 081
EPXF 016    ECOL 145
EPXF 020    EPXF 017
EPXF 024    EPXF 009
EPXF 039    EPXF 068
EPXF 041    EPXF 023
EPXF 049    EPXF 047
EPXF 055    EPXF 048
EPXF 057    EPXF 055
EPXF 059    EPXF 057
EPXF 061    EPXF 059
EPXF 063    EPXF 061
EPXF 065    EPXF 063
EPXF 069    ECOL 144  EPXF 067
```

ERCX DESCRIPTIVE TEXT

ENTRY POINTS

START — ENTRY IS MADE FROM THE ERDR ROUTINE TO START READER COLUMN IMAGE TO STORAGE TRANSFER. SEE FEMDM DIAGRAM 5-105.

WLRSET — ENTRY IS MADE HERE FROM THE ERDR ROUTINE TO SET THE WLR INDICATOR WHEN THE WLR COUNTER HAS OVERFLOWED DURING A READ OR SENSE COMMAND WITH THE SKIP FLAG ON.

CDFLAG — ENTRY IS FROM THE ERDR ROUTINE WHEN THE SKIP OPERATION OF A READ OR PFR READ IS COMPLETED, (CCW COUNT=0). IF NOT DATA CHAINING, CONTINUE WITH THE NORMAL ENDING PROCEDURE. IF DATA CHAINING PREPARE TO FETCH A NEW CCW.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERCX 001	T			XFER FROM COLUMN FORM BUFFER(RO) TO CCW ADDR	
1D84	5E88	ERCX 002	READ			RDH H AS,I+2	RD 2 BYTES
1D86	6FF3	ERCX 003				H1=H1+H1	
1D88	6FF3	ERCX 004				H1=H1+H1	SHIFT 8,9 TWICE
1D8A	5FB3	ERCX 005				T1=H1XH	T1=8,9
1D8C	6BE5	ERCX 006				T1=T1$H0	T1=8,9,12,11,0,1,2,3
1D8E	1B73	ERCX 007				T1=T1*-K07	T1=8,9,12,11,0
1D90	5FF9	ERCX 008				H1=H1	H1=4,5,6,7,8,9
1D92	E096	ERCX 009		011	NOHI	BR IF HZNZ	BR IF 4,5,6,7 NO BIT
1D94	9A71	ERCX 010		C55	CHECK N	N=H0 BITS567	MASK ON BITS 1,2,3
1D96	0F83	ERCX 011	NOHI			Z=H0*-K08	BITS 1,2,3
1D98	F0A4	ERCX 012		018	GOVDTY	BR IF LZNZ	BR IF ANY IN 1,2,3
1D9A	3B43	ERCX 013				T1=T1$K04	AT LEASE 4
1D9C	CF21	ERCX 014		016	BIT4ON	BR IF H1 BIT0=1	BR IF BIT 4
1D9E	9F71	ERCX 015		055	CHECK N	N=H1 BITS123	MASK ON BITS 5,6,7
1DA0	0F85	ERCX 016	BIT4ON			Z=H1*-K80	4 BIT ON
1DA2	E0CD	ERCX 017		021	XLATE	BR IF HZ=0	BR IF NO OTHER BIT
1DA4	98EE	ERCX 018	GOVDTY	065	CHECK 7	BR	
1DC8	3B23	ERCX 019	ORIN2			T1=T1$K02	OR IN 2 BIT
1DCA	3B13	ERCX 020	ORIN1			T1=T1$K01	OR IN 1 BIT
1DCC	5FA0	ERCX 021	XLATE			RDB H1 AS,T+0	XLATE TABLE LOOK UP
1DCE	7F38	ERCX 022				STB H1 V+1	SEND TO MAIN STORE
1DD0	21FF	ERCX 023	DCRCT			U1=U1+KFF	DECREMENT CCW CT
1DD2	C4A9	ERCX 024		032	CDFLG	BR IF Z=0	
1DD4	261B	ERCX 025	START			D0=D0+K01	WLR CTR + 1
1DD6	F4A7	ERCX 026		031	WLRSET	BR IF AC=1	BR IF WLR
1DD8	E404	ERCX 027		002	READ	BR IF GO BIT2=0	BR IF NOT COL BIN
1DDA	5F88	ERCX 028				RDB H1 AS,I+1	COL BIN
1DDC	7F38	ERCX 029				STB H1 V+1	MOVE TO MAIN STORE
1DDE	9DD0	ERCX 030		023	DCRCT	BR	
1DA6	3C13	ERCX 031	WLRSET			P0=P0$K01	WLR INDICATOR
1DA8	5629	ERCX 032	CDFLG			V0=D0	MOVE WLR CTR
1DAA	2607	ERCX 033				D0=0	RST D0
1DAC	54E9	ERCX 034				H0=G0	MOVE CMD BACK TO GO
1DAE	5460	ERCX 035				RDH G AS,D+0	RESTORE CHNL STATUS
1DB0	C536	ERCX 036		039	CDANOT	BR IF G1 BIT0=0	BR IF NOT CDA
1DB2	F83D	ERCX 037		042	PFRBRK	BR IF P0 BIT7=1	BR IF WLR

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
1DB4     9B90     ERCX 038              ECOL 069  GOCHAN     BR                     CDA
1DB6     221B     ERCX 039  CDANOT                         V0=V0+K01              INCREMENT WLR CTR
1DB8     F4BD     ERCX 040                    042  PFRBRK     BR IF AC=1
1DBA     3C13     ERCX 041                                   P0=P0$K01              WLR
1DBC     71A2     ERCX 042  PFRBRK                           STB U1 DA,AC           STORE RESIDUE CT
1DBE     F701     ERCX 044                    049  ENPFR      BR IF D1 BIT3=1        BR IF NOT PFR
1DC0     CBC4     ERCX 045                    047  NOVAL      BR IF PS BIT4=0        PFR READ,BR IF NO VAL CHK(RU031)
1DC2     3D83     ERCX 046                                   P1=P1$K08              VAL CHK,OR BIT IN SENSE
1DC4     0D8C     ERCX 047  NOVAL                            RST RP K=08            RST PCH CMD INTERLOCK
1DC6     9212     ERCX 048              EXFR 118  COMBIN     BR
1DC8     B3E8     ERCX 049  ENPFR       ERRQ 101  STIND      BAL                    STORE INDICATOR AND SENSE
1DC2     998C     ERCX 053              ERDR 114  STACKR     BR
18DE     3B23     ERCX 054  ORBIT2                          T1=T1$K02              OR IN 2 BIT
18E0     9DCC     ERCX 055  CHECK  0          021  XLATE      BR
18E2     9DC8     ERCX 056  CHECK  1          019  ORIN2      BR                     3 OR 7 BIT
18E4     98DE     ERCX 057  CHECK  2          054  ORBIT2     BR                     2 OR 6 BIT
18E6     98EE     ERCX 058  CHECK  3          065  CHECK  7   BR
18E8     9DCA     ERCX 059  CHECK  4          020  ORIN1      BR                     1 OR 5 BIT
18EA     98EE     ERCX 060  CHECK  5          065  CHECK  7   BR
18EC     98EE     ERCX 061  CHECK  6          065  CHECK  7   BR
18EE     F775     ERCX 065  CHECK  7          068  RDVDCK     BR IF D1 BIT3=1        VALIDITY CHECK
18F0     2F80     ERCX 066                                   SET P K=08             SET PFR VAL CHK(RU031)
18F2     9DCC     ERCX 067                    021  XLATE      BR
18F4     2B80     ERCX 068  RDVDCK                           SET R K=08             SET RDR VAL CHK(RU023)
18F6     9DCC     ERCX 070                    021  XLATE      BR
                  ERCX 071  *                                NATIVE 2540 READER TRANSLATE TABLE
                  ERCX 072  ATABLE  ADDR=3000
3000     40F1     ERCX 073  C                                XAUX'40F1F2F3F4F5F6F7F061E2E3E4E5E6E760D1D2D3D4D5'
3016     D6D7     ERCX 074  C                                XAUX'D6D7D0A1A2A3A4A5A6A750C1C2C3C4C5C6C7C0818283'
302C     8485     ERCX 075  C                                XAUX'848586876A91929394959697708182B3B4B5B6B7F931'
3042     3233     ERCX 076  C                                XAUX'323334353637E92122232425262728D9111213141516 17'
3058     A9E1     ERCX 077  C                                XAUX'A9E1626364656667C9C102030405060708941424344 45'
306E     4647     ERCX 078  C                                XAUX'4647995152535455565 7B971727374757677F8797A7B'
3084     7C7D     ERCX 079  C                                XAUX'7C7D7E7FE869E06B6C6D6E6FD8595A5B5C5D5E5FA8A0'
309A     AAAB     ERCX 080  C                                XAUX'AAABACADAEAFC8494A4B4C4D4E4F88808A8B8C8D8E8F'
30B0     9890     ERCX 081  C                                XAUX'98909A9B9C9D9E9FB8B0BABBBCBDBEBF38393A3B3C3D'
30C6     3E3F     ERCX 082  C                                XAUX'3E3F28292A2B2C2D2E2F18191A1B1C1D1E1F6820EAEB'
30DC     ECED     ERCX 083  C                                XAUX'ECEDEEEF08090A0B0C0D0E0F4800CACBCCCDCECF5810'
30F2     DADB     ERCX 084  C                                XAUX'DADBDCDDDEDF7830FAFBFCFDFEFF'
                  ERCX 085  AEND

                                        **********************************
                                        * CROSS REFERENCE FOR CSECT ERCX *
                                        **********************************

ERCX 002    ERCX 027
ERCX 011    ERCX 009
ERCX 016    ERCX 014
ERCX 018    ERCX 012
ERCX 019    ERCX 056
ERCX 020    ERCX 059
ERCX 021    ERCX 017  ERCX 055  ERCX 067  ERCX 070
ERCX 023    ERCX 030
```

****
* CROSS REFERENCE FOR CSECT ERCX *
****

```
ERCX 025    ERDR 169
ERCX 031    ERCX 026  ERDR 172
ERCX 032    ERCX 024  ERDR 178
ERCX 039    ERCX 036
ERCX 042    ERCX 037  ERCX 040
ERCX 047    ERCX 045
ERCX 049    ERCX 044
ERCX 054    ERCX 057
ERCX 055    ERCX 010  ERCX 015
ERCX 065    ERCX 018  ERCX 058  ERCX 060  ERCX 061
ERCX 068    ERCX 065
```

ENTRY POINTS

NRDR	ENTRY TO THIS LABEL IS FROM THE DCLB ROUTINE FOR A READER OR PUNCH START I/O, OR FOR A CHAINING OPERATION. A CHECK IS MADE FOR IPL, AND THE AVAILABILITY OF THE READER OR THE PUNCH.
AVANOT	ENTRY IS MADE HERE FROM THE ERRQ ROUTINE WHEN THE 2540 IS FOUND NOT AVAILABLE. CONDITION CODE 3 IS SET, THE ACTIVE AND SECONDARY BITS ARE RESET AND THE DPTQ ROUTINE IS BRANCHED TO FOR A UCW STORE.
STACKR	ENTRY IS MADE HERE FROM THE ERCX ROUTINE TO SET STACKER SELECT BITS, SET NOT-READ, AND STORE INDICATOR AND SENSE BYTES.
SSDONE	ENTRY IS MADE TO THIS LABEL FROM THE ERRQ ROUTINE TO SET THE FEED INDICATOR, AND SET UP AND ISSUE A FEED COMMAND WITH STACKER SELECT.
UEXPFR	ENTRY IS FROM THE EPCH ROUTINE WHEN A PFR UNIT EXCEPTION IS DETECTED. SET UNIT EXCEPTION IN STATUS REGISTER AND RESET COMMAND INTERLOCK. IF IPL, GO TO THE ERRQ ROUTINE AND STOP. IF NOT IPL, GO TO THE DPTS ROUTINE TO STORE STATUS AND FLAGS IN THE UCW.
CMDREJ	ENTRY IS MADE HERE FROM THE EPCH ROUTINE WHEN AN INVALID COMMAND DECODE, OR AN INVALID COMMAND SEQUENCE IS DETECTED.
BACK	THE EPCH ROUTINE BRANCHES HERE TO SETUP THE BEGINNING ADDRESSES OF THE COLUMN IMAGE BUFFER AND TRANSLATE TABLE MODULE.
BRSKIP	ENTRY HERE IS FROM THE EXFR ROUTINE AFTER THE PFR COLUMN BUFFER ADDRESS IS SETUP. IF THE SKIP FLAG IS ON, DO SKIP INDICATED. IF SKIP FLAG OFF, GO TO THE ERCX ROUTINE TO START TRANSLATE OF COLUMN IMAGE TO BYTE CHARACTERS FOR PROGRAM STORAGE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ERDR 001	T				NATIVE 2540 RDR OR PCH SIO OR CHAIN DECODE ROUTINE
		ERDR 002	*				
		ERDR 003	*				ENTRY TO THIS ROUTINE IS MADE FROM DCLB TO LABEL NRDR OF THIS ROUTINE.
		ERDR 004	*				THE REGISTERS CONTAIN THE FOLLOWING INFO
		ERDR 005	*				
		ERDR 006	*				COUNT U
		ERDR 007	*				DATA ADDRESS V
		ERDR 008	*				COMMAND BYTE G0
		ERDR 009	*				FLAG G1
		ERDR 010	*				
		ERDR 011	*				NATIVE 2540 READER UCW FORMAT(AUX STORAGE)
		ERDR 012	*				
		ERDR 013	*				00D8 00DA 00DC 00DE
		ERDR 014	*				*----------*----------*----------*----------*
		ERDR 015	*				* * * * *
		ERDR 016	*				* CANNEL *CHANNEL *UNIT *NOT * RESIDUAL * NEXT CCW *
		ERDR 017	*				* STATUS* FLAGS * STATUS * USED * COUNT * ADDRESS *
		ERDR 018	*				* * * * * * *
		ERDR 019	*				*----------*----------*----------*----------*----------*----------*
		ERDR 020	*				
		ERDR 021	*				
		ERDR 022	*				
		ERDR 023	*				

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT                COMMENTS

                ERDR 024    *                    CH STAT-D8      *  CH FLAGS-D9     *  UNIT STAT-DA
                ERDR 025    *                    0-SECONDARY     *  0-CHAIN DATA    *  0-NOT USED
                ERDR 026    *                    1-WLR           *  1-CHAIN CMD     *  1-NOT USED
                ERDR 027    *                    2-PROG CHK      *  2-SLI           *  2-NOT USED
                ERDR 028    *                    3-PROT CHK      *  3-SKIP          *  3-BUSY
                ERDR 029    *                    4-NOT USED      *  4-PCI           *  4-CH END
                ERDR 030    *                    5-NOT USED      *  5-ACTIVE        *  5-DEV END
                ERDR 031    *                    6-INTFCE CK     *  6-NOT USED      *  6-UNIT CHK
                ERDR 032    *                    7-INT IN BFR    *  7-NOT USED      *  7-UNIT EXCEPT
                ERDR 033    *
                ERDR 034    *                          **********************************
                ERDR 035    *                          *  INDICATORS  AND SENSE BYTE  *
                ERDR 036    *                          **********************************
                ERDR 037    *                    *-----------------------------------------*
                ERDR 038    *                    *  INDICATORS           *  SENSE           *
                ERDR 039    *                    *  AUX STORAGE          *  AUX STORAGE     *
                ERDR 040    *                    *  ADDRESS              *  ADDRESS         *
                ERDR 041    *                    *  10F0                 *  10F1            *
                ERDR 042    *                    *-----------------------*------------------*
                ERDR 043    *                    * 0-CHAIN DATA         *  0- CMD REJECT   *
                ERDR 044    *                    * 1-STACKER SEL        *  1- NOT READY    *
                ERDR 045    *                    * 2-STACKER SEL        *  2- NOT USED     *
                ERDR 046    *                    * 3-0                  *  3- EQUIP CHK    *
                ERDR 047    *                    * 4-READ               *  4-VALIDITY CHK  *
                ERDR 048    *                    * 5-COUNT STORED       *  5- NOT USED     *
                ERDR 049    *                    * 6-FEED               *  6-UNUSUAL CMD   *
                ERDR 050    *                    * 7-WLR                *  7- NOT USED     *
                ERDR 051    *                    * ---------------------------------------*
                ERDR 052    ASEQ    AL07=12
1912    54E9    ERDR 053     NRDR                                  H0=G0                  MOVE CMD INTO H0
1914    C4A0    ERDR 054                          060   NOTCDA    BR IF ZNZ              BR IF NOT DATA CHAIN
1916    5E82    ERDR 055                                          RDH H DA,AE            CDA, RESTORE OLD CMD INTO H0
1918    5D69    ERDR 056                                          D0=P1                  RESTORE WLR CTR
191A    8A68    ERDR 057                ERRQ 091  RDIND     BAL                    READ INDICATORS & SENSE INTO P
191C    3C85    ERDR 058                                          P0=P0$K80              SET CDA INDICATOR
191E    950E    ERDR 059                EPCH 058  HIGHCT    BR                     GO TO ADJUST HIGH COUNT
1920    C3AE    ERDR 060    NOTCDA            067   NOTIPL    BR IF BA BIT4=0        BR IF NOT IPL
1922    F72C    ERDR 061                      066   GOSTOP    BR IF D1 BIT3=0        BR IF IPL FROM PUNCH
1924    2406    ERDR 062                                          SET MODE K=30          SET READER MODE
1926    CEAC    ERDR 063                      066   GOSTOP    BR IF RPS BIT0=0       BR IF OFF LINE(SX041)
1928    3D23    ERDR 064                                          P1=P1$K02              SET WAIT BIT FOR IPL CLUTCHING
192A    DDAF    ERDR 065                      067   NOTIPL    BR IF RS BIT1=1        BR IF RDR READY(SX041)
192C    9F5C    ERDR 066    GOSTOP    ERRQ 049  STOP      BR    **IPL NOT AVAILABLE OR NOT READY,OR IPL FROM PCH
192E    9812    ERDR 067    NOTIPL    CCOM 056  LSAVE     BAL                    SIO OR CC, SAVE U, G, I & P
1930    8A68    ERDR 068              ERRQ 091  RDIND     BAL                    P0=INDICATORS, P1=SENSE
1932    CEBD    ERDR 069                      074   AVABLE    BR IF RPS BIT0=1       BR IF ON LINE(SX041)
1934    8178    ERDR 070              CCOM 066  LRSTRB    BAL                    RESTORE P AND I
1936    2C75    ERDR 071    AVANOT                                P0=0$K70               NOT AVAILABLE, SET COND. CODE 3
1938    4452    ERDR 072                                          RDH G DC,9A            G=0000, RST ACTIVE, SECONDARY
193A    8D6A    ERDR 073              DPTR 076  STINST    BR                     STORE UCW & GO TO BSWI
193C    5460    ERDR 074    AVABLE                                RDH G AS,D+0           RD CH.ST. SO LATER CAN STORE UCW
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
193E  F747  ERDR 075                 079    RDR1       BR IF D1 BIT3=1    BR IF RDR
1940  2D80  ERDR 076                                    SET RP K=08        SET PCH CMD INLK(RU031)
1942  0F30  ERDR 077                                    RST P K=03         RST PCH REQUEST IF HWDE(RU015)
1944  994A  ERDR 078                 081    H10A       BR
1946  2D04  ERDR 079  RDR1                              SET RP K=20        SET RDR CMD INLK(RU021)
1948  0B30  ERDR 080                                    RST R K=03         RST RDR REQUEST IF HWDE(RU015)
194A  2FA3  ERDR 081  H10A                               H1=0$K0A         SET H1 TO LATER RST INDICATORS
194C  0E3B  ERDR 082                                     Z=H0<K03         MASK FOR NO OP
194E  C4DC  ERDR 083                 090    ENNOOP     BR IF ZNZ          BR IF NOT NO OP
1950  F758  ERDR 084                 088    PCH1       BR IF D1 BIT3=0    NO OP, BR IF PCH
1952  DDDA  ERDR 085                 089    NOTRDY     BR IF RS BIT1=0    BR IF RDR NOT READY(SX041)
1954  B43C  ERDR 086  RDYNOP  ERRQ 109    GUESS      BAL                RST CMD INTERLOCK
1956  8C58  ERDR 087           DPTS 125    NOOP       BR                 SHARE 1403 NO OP ROUTINE
1958  CFD5  ERDR 088  PCH1              086    RDYNOP     BR IF PS BIT0=1    BR IF PCH READY(SX041)
195A  99B6  ERDR 089  NOTRDY            139    UNITCK     BR                 NOT READY
195C  0E4B  ERDR 090  ENNOOP                               Z=H0<K04         MASK FOR SENSE CMD
195E  F76F  ERDR 091                 099    RDR2       BR IF D1 BIT3=1    BR IF RDR
1960  F0E9  ERDR 092                 096    PCHSNS     BR IF LZ=0         BR IF PCH SENSE CMD
1962  2D07  ERDR 093                                    P1=0               NOT SENSE CMD, ZERO OUT SENSE
1964  CFDA  ERDR 094                 089    NOTRDY     BR IF PS BIT0=0    BR IF PCH NOT READY(SX041)
1966  94D4  ERDR 095           EPCH 010    NPCH       BR                 GO TO PCH CMD DECODE
1968  CFF3  ERDR 096  PCHSNS            101    SNSRDY     BR IF PS BIT0=1    BR IF PCH READY(SX041)
196A  3D45  ERDR 097  SNSNRY                               P1=P1$K40        SENSE CMD, NOT READY
196C  9974  ERDR 098                 102    JOIN       BR
196E  F0FA  ERDR 099  RDR2            105    NOTSNS     BR IF LZNZ         BR IF NOT SENSE CMD
1970  DDEA  ERDR 100                 097    SNSNRY     BR IF RS BIT1=0    RDR SNS CMD,BR IF NOT RDY(SX041)
1972  1D45  ERDR 101  SNSRDY                             P1=P1*-K40       SENSE CMD, READY
1974  B43C  ERDR 102  JOIN    ERRQ 109    GUESS      BAL                RST CMD INTERLOCK
1976  3623  ERDR 103                                     D0=D0$K02        SET 6 BIT FOR 1403 TO BR ON
1978  A5E0  ERDR 104           DYPE 176    GOCAL      BR                 GO TO PICK UP H1=P1 WORD
197A  2D07  ERDR 105  NOTSNS                              P1=0             RDR. NOT SNS CMD, ZERO OUT SNS
197C  DDDA  ERDR 106                 089    NOTRDY     BR IF RS BIT1=0    BR IF NOT RDY(SX041)
197E  1B00  ERDR 107                                     RST R K=80       RST RD & VAL CHK(RU023)
1980  6CF7  ERDR 108                                     P0=P0*H1         RST ALL IND. EXCEPT FD IND.
1982  EA34  ERDR 109                 138    CMDREJ     BR IF H0 BIT6=0    BR IF NO 6 BIT - CMD REJECT
1984  FA26  ERDR 110                 127    RDORRF     BR IF H0 BIT7=0    BR IF RD OR RD-FD
1986  EE34  ERDR 111                 138    CMDREJ     BR IF H0 BIT2=0    CMD REJ IF NO 2 BIT
1988  E835  ERDR 112                 138    CMDREJ     BR IF P0 BIT6=1    FEED CMD, BR IF DOUBLE FD
198A  76F2  ERDR 113                                     STH D DA,BE      FD ONLY, IMMEDIATE CMD,SAVE D07
198C  CE1A  ERDR 114  STACKR            121    SS1OR2     BR IF H0 BIT0=0    BR IF STKR SEL 1 OR 2
198E  DE18  ERDR 115                 120    RP3        BR IF H0 BIT1=0    BR IF RP3
1990  FA35  ERDR 116                 138    CMDREJ     BR IF H0 BIT7=1    BR IF NOT RD, CMD REJECT
1992  0D04  ERDR 117                                    RST RP K=20        RST RDR CMD INLK(RU021)
1994  8F7A  ERDR 118           ERRQ 082    RDCHST     BAL                RESTORE CH. STATUS AND FLAGS
1996  920A  ERDR 119           EXFR 114    CHKCHK     BR                 RD NO FD, FROM ERCX, NOT PFR
1998  3C45  ERDR 120  RP3                                P0=P0$K40        STKR SELECT RP3
199A  DE1E  ERDR 121  SS1OR2            123    SSDONE     BR IF H0 BIT1=0    BR IF R1 OR RP3
199C  3C25  ERDR 122                                     P0=P0$K20        STKR SELECT R2
199E  1C83  ERDR 123  SSDONE                             P0=P0*-K08       RST READ INDICATOR
19A0  3C23  ERDR 124                                     P0=P0$K02        SET FEED INDICATOR
19A2  B3E8  ERDR 125           ERRQ 101    STIND      BAL                STORE INDICATORS & SENSE
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
19A4   9C05    ERDR 126                  180    SETSKR N    N=P0 BITS123          ISSUE FD CMD & STKR SELECT
19A6   DA35    ERDR 127    RDORRF        138    CMDREJ     BR IF H0 BIT5=1       CMD REJECT IF 5-6 COMBINATION
19A8   EDAF    ERDR 128                  131    UNITEX     BR IF R5 BIT2=1       BR IF UNIT EXCEPTION(SX042)
19AA   2B40    ERDR 129                                     SET R K=04            RD OR RD-FD CMD,SENSE SET(RU019)
19AC   9504    ERDR 130                EPCH 053  PICKUP     BR          **SHARE UNUSUAL CMD SEQ BR & COUNT ADJUST
19AE   3D00    ERDR 131    UNITEX                           SET RP K=80           RST EOF LITE(RU023)
19B0   2E13    ERDR 133    UEXPER                           H0=0$K01              UNIT EXCEPTION IN UNIT STAT
19B2   99B8    ERDR 137                  140    GOSTIN     BR
19B4   2D85    ERDR 138    CMDREJ                           P1=0$K80              CMD REJ IN SENSE
19B6   2E23    ERDR 139    UNITCK                           H0=0$K02              UNIT CK IN UNIT STAT OF CSW
19B8   B3E8    ERDR 140    GOSTIN    ERRQ 101  STIND      BAL                   STORE P - IND, & SENSE
19BA   B43C    ERDR 141              ERRQ 109  GUESS      BAL                   RST CMD INTERLOCK
19BC   2A07    ERDR 142                                     T0=0                  RST TO IN CASE OF CDA FLAG ON
19BE   C3C2    ERDR 143                  145    IPLNOT     BR IF BA BIT4=0       BR IF NOT IPL
19C0   9F5C    ERDR 144              ERRQ 049  STOP       BR                    IPL CMD REJ OR UNIT EXCEPTION
19C2   8C60    ERDR 145    IPLNOT    DPTS 137  PCENTR     BR
19C4   CC5D    ERDR 146    BACK          162    MOVEOP     BR IF P0 BIT0=1       PICKED UP CDA & PFR RD,BR IF CDA
19C6   F753    ERDR 148                  154    ENPFR      BR IF D1 BIT3=1       SIO OR CC, BR IF NOT PFR-RD
19C8   2445    ERDR 149                                     G0=0$K40              PFR-RD SIO OR CC
19CA   2513    ERDR 150                                     G1=0$K01              G=4001 TO PFR COL BUFR
19CC   40A6    ERDR 151                                     U=T                   U=10F2
19CE   31BB    ERDR 152                                     U1=U1-K0B   **U=10E6  FOR LAST 10 BYTES IN IMAGE BUFR
19D0   915A    ERDR 153              EXFR 012  BYTECT      BR                    PFR-RD. XFER FIRST & THEN RD
19D2   EE56    ERDR 154    ENPFR         159    NOTCOL     BR IF HC BIT2=0       BR IF NOT COL BIN
19D4   36A9    ERDR 158                                     D0=0-KA0              COL BIN. WLR CTR=5F
19D6   2855    ERDR 159    NOTCOL                           I0=0$K50
19D8   2915    ERDR 160                                     I1=0$K10
19DA   3983    ERDR 161                                     I1=I1$K08             I=5018, COL FORM BUFR ADDR
19DC   5E49    ERDR 162    MOVEOP                           G0=H0                 MOVE CMD TO GO FOR ERCX ROUTINE
19DE   F565    ERDR 164    BRSKIP        170    SKIP       BR IF G1 BIT3=1       BR IF SKIP FLAG ON
19E0   2A35    ERDR 168                                     T0=0$K30              XLATE TABLE ADDR
19E2   9DD4    ERDR 169              ERCX 025  START      BR    **GO TO RD, COL OR NORMAL,OR PFR.SIO,CC,OR CDA
19E4   261B    ERDR 170    SKIP                             D0=D0+K01             WLR CTR + 1
19E6   F4EA    ERDR 171                  173    IADD1      BR IF AC=0            BR IF NO OVER CT
19E8   9DA6    ERDR 172              ERCX 031  WLRSET     BR                    SKIPPED TOO MANY, WLR
19EA   5884    ERDR 173    IADD1                            I=I+1                 ADD 1 TO COL BUFR ADDR
19EC   E471    ERDR 174                  176    CTADD1     BR IF G0 BIT2=1       BR IF COL BIN
19EE   5884    ERDR 175                                     I=I+1                 TWO BYTES OF COL FORM PER BYTE
19F0   21FF    ERDR 176    CTADD1                           U1=U1+KFF             ADJUSTED CCW CT LESS 1
19F2   C4E4    ERDR 177                  170    SKIP       BR IF ZNZ             BR IF NOT DONE SKIPPING
19F4   9DA8    ERDR 178              ERCX 032  CDFLG      BR                    FINISH SKIPPING
               ERDR 179    AEND
1A00   2B08    ERDR 180    SETSKR 0                         SET R K=40            PKT R1(RU021)
1A02   9150    ERDR 181              EXFR 007  XFER       BR                    GO TO XFER ROW FORM TO COL BUFR
1A04   2B0C    ERDR 182    SETSKR 2                         SET R K=60            PKT R2(RU021 & 023)
1A06   9150    ERDR 183              EXFR 007  XFER       BR
1A08   2B0A    ERDR 184    SETSKR 4                         SET R K=50            PKT RP3(RU021 & 023)
1A0A   9150    ERDR 185              EXFR 007  XFER       BR
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT ERDR *
                                  **********************************
ERDR 053     DCLB 247  DCLB 248
ERDR 060     ERDR 054
ERDR 066     ERDR 061  ERDR 063
ERDR 067     ERDR 060  ERDR 065
ERDR 071     ERRQ 007
ERDR 074     ERDR 069
ERDR 079     ERDR 075
ERDR 081     ERDR 078
ERDR 086     ERDR 088
ERDR 088     ERDR 084
ERDR 089     ERDR 085  ERDR 094  ERDR 106
ERDR 090     ERDR 083
ERDR 096     ERDR 092
ERDR 097     ERDR 100
ERDR 099     ERDR 091
ERDR 101     ERDR 096
ERDR 102     ERDR 098
ERDR 105     ERDR 099
ERDR 114     ERCX 053
ERDR 120     ERDR 115
ERDR 121     ERDR 114
ERDR 123     ERDR 121  ERRQ 068
ERDR 127     ERDR 110
ERDR 131     ERDR 128
ERDR 133     EPCH 051
ERDR 138     EPCH 032  ERDR 109  ERDR 111  ERDR 112  ERDR 116  ERDR 127
ERDR 139     ERDR 089
ERDR 140     ERDR 137
ERDR 145     ERDR 143
ERDR 146     EPCH 072
ERDR 154     ERDR 148
ERDR 159     ERDR 154
ERDR 162     ERDR 146
ERDR 164     EXER 099
ERDR 170     ERDR 164  ERDR 177
ERDR 173     ERDR 171
ERDR 176     ERDR 174
ERDR 180     ERDR 126
```

ERRQ DESCRIPTIVE TEXT

ENTRY POINTS

REDUC — ENTRY IS FROM THE DCLA ROUTINE WHEN A START I/O OR TEST I/O IS DECODED (FOR READER OR PUNCH), AND DEVICE END HAS BEEN CLEARED.

PCHST — ENTRY IS FROM THE BSWI ROUTINE WHEN A PUNCH STATUS REQUEST HAS BEEN DETECTED. READOUT THE UCW BYTES INTO LOCAL STORAGE, RESET READER STATUS AND DEVICE END.

RDST — ENTRY IS FROM THE BSWI ROUTINE WHEN A READER STATUS REQUEST HAS BEEN DETECTED. READOUT THE UCW BYTES INTO LOCAL STORAGE, RESET READER STATUS AND DEVICE END.

STOP — ENTRY IS FROM THE DCLR ROUTINE WHEN A CHANNEL ERROR IS DETECTED DURING AN IPL. ENTRY IS FROM THE ERDR ROUTINE WHEN THE 2540 IS NOT AVAILABLE FOR IPL, NOT READY, IPL FROM PUNCH ATTEMPTED, IPL COMMAND REJECT, OR A UNIT EXCEPTION DETECTED.

RDCONT, RDCHST, RPHUCW, RDIND, STIND, GUESS — THESE ARE COMMON ENTRY POINTS BRANCHED TO BY BRANCH AND LINK WORDS TO PERFORM SOME OF THE BASIC FUNCTIONS FOR THE 2540.

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS

            ERRQ 001   T                          2540 REQUEST & TEST I/O ROUTINE
            ERRQ 002   *
1F24  8F70  ERRQ 004    REDUC          C76  RDUCW     BAL                  READ CH.STAT, FLAGS, & UNIT STAT
1F26  3623  ERRQ 005                                  D0=D0$K02            SET D0 6 BIT TO BE BR ON LATER
1F28  CEAD  ERRQ 006                          008  AVABLE    BR IF RPS BIT0=1     BR IF AVABLE(SX041)
1F2A  9936  ERRQ 007                    ERDR  071  AVANOT    BR                   NOT AVAILABLE
1F2C  F710  ERRQ 008   AVABLE          013  PCH1      BR IF D1 BIT3=0      BR IF PCH
1F2E  C439  ERRQ 009                   031  PCKUPD    BR IF GO BIT0=1      BR IF SECONDARY ON, RDR
1F30  DD94  ERRQ 010                   015  NOTRDY    BR IF RS BIT1=0      BR IF RDR NOT RDY(SX041)
1F32  D995  ERRQ 011                   015  NOTRDY    BR IF RS BIT5=1      BR IF HARDWARE DEVICE END ON
1F34  810C  ERRQ 012   RDY       CCOM  155  CCOB      BR                   BR TO SET COND. CODE 0
1F10  C46B  ERRQ 013   PCH1            018  PICKUP    BR IF GO BIT0=1      BR IF SECONDARY ON, PCH
1F12  CFB5  ERRQ 014                   012  RDY       BR IF PS BIT0=1      BR IF PCH RDY(SX041)
1F14  8CD4  ERRQ 015   NOTRDY    DPTT  013  UCENT     BR                   NOT READY
            ERRQ 016   *                        2540 REQUEST
1F68  8F64  ERRQ 017   PCHST           086  RPHUCW    BAL            **PCH REQ. RD CH.ST,FLAG,UNIT STAT, CT
1F6A  DB96  ERRQ 018   PICKUP         055  NODE      BR IF PS BIT5=0      BR IF NO PCH HWDE(SX043)
1F6C  0F10  ERRQ 019                                  RST P K=01           RST PCH HWDE(RUC15)
1F6E  C442  ERRQ 020                   036  SETSEC    BR IF GO BIT0=0      BR IF NOT-READY TO READY
1F70  FFC4  ERRQ 021                   037  CCBR      BR IF PS BIT3=0      BR IF NO PCH CHK (SX042)
1F72  9816  ERRQ 022                CCOM  058  LSAVEB    BAL                  PCH CK, SAVE P, I
1F74  2D15  ERRQ 023                                  P1=0$K10             EQUIPMENT CK IN SENSE
1F76  5CCD  ERRQ 024                                  P0=P0L               RETAIN PFR RD & PFR WR INDICATOR
1F78  B3E8  ERRQ 025                   101  STIND     BAL                  STORE INDICATORS & SENSE
1F7A  7CA0  ERRQ 026                                  STH P AS,T+0         STORE FORCED STKR SEL IN 1OF2
1F7C  1545  ERRQ 027                                  G1=G1*-K40           BREAK CC
1F7E  3E23  ERRQ 028                                  H0=H0$K02            UNIT CK, DE IN UNIT STAT
1F80  9F44  ERRQ 029                   037  CCBR      BR
1F36  8F6C  ERRQ 030   RDST            074  RRDUCW    BAL            **RDR REQ. RD CH.ST,FLAG,UNIT STAT, CT
1F38  2466  ERRQ 031   PCKUPD                       SET MODE K=36        RDR MODE, 2540 ZONE
1F3A  E607  ERRQ 032                   064  CPUXFR    BR IF D0 BIT2=1      BR IF RUN-IN CPU XFER
1F3C  2406  ERRQ 033                                  SET MODE K=30        SET BACK FIRST TO RDR MODE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
1F3E    D996    ERRQ 034                 055     NODE       BR IF RS BIT5=0    BR IF NO RDR HWDE(SX043)
1F40    0B10    ERRQ 035                                    RST R K=01         RST RDR HWDE
1F42    3485    ERRQ 036    SETSEC                          GO=GO$K80    **SET SECONDARY IN CASE OF QUE HWDE-READY
1F44    8FCE    ERRQ 037    CCBR     DCOM 025  RSTREQ     BAL                RST RDR OR PCH REQUEST
1F46    D54A    ERRQ 038                 040     NOCC       BR IF G1 BIT1=0    BR IF NO CMD CHAIN
1F48    8D64    ERRQ 039                 DPTR 073  CHAIOK   BR                 GO TO CHAIN
1F4A    3E43    ERRQ 040    NOCC                            HO=HO$K04          DE IN UNIT STAT
1F4C    E261    ERRQ 041                 050     TIO        BR IF DO BIT6=1    BR IF FROM TIO OR SIO
1F4E    F75E    ERRQ 042                 063     GOINT      BR IF D1 BIT3=0    BR IF PCH
1F50    C3DE    ERRQ 043                 063     GOINT      BR IF BA BIT4=0    BR IF NOT IPL
1F52    D45D    ERRQ 044                 049     STOP       BR IF GO BIT1=1    IPL, BR IF WLR IN CH. STAT
1F54    EA5D    ERRQ 045                 049     STOP       BR IF HO BIT6=1    BR IF UNIT CHECK
1F56    4452    ERRQ 046                                    RDH G DC,9A        G=0000, RST ACTIVE, SECONDARY
1F58    B406    ERRQ 047                 DCOM 081  REDSTR    BAL                STORE UCW
1F5A    A7F2    ERRQ 048                 DCLL 040  PICKUP    BR                 IPL DONE
1F5C    A7D8    ERRQ 049    STOP         DCLL 026  STOP      BR                 BR TO IPL STOP
1F60    2A07    ERRQ 050    TIO                             TO=0               RST TO
1F62    8FCE    ERRQ 051                 DCOM 025  RSTREQ     BAL                RST READER,PCH REQUEST
1F64    D104    ERRQ 052                 054     NOACT      BR IF G1 BIT5=0    BR IF NO ACTIVE BIT
1F66    8CDE    ERRQ 053                 DPTT 023  RESTES     BR                 TIO OR SIO, ACT,SEC=11
1F04    8CA4    ERRQ 054    NOACT        DPTT 049  SHARE      BR                 TIO, ACT,SEC=01
1F16    E261    ERRQ 055    NODE                    050     TIO        BR IF DO BIT6=1    BR IF TIO OR SIO
1F18    C3C5    ERRQ 056                 037     CCBR       BR IF BA BIT4=1    BR IF IPL
                ERRQ 057  *                                    INTERFACE CHECK OR UNQUEUED
1F1A    C422    ERRQ 058                 062     INTFCK     BR IF GO BIT0=0    BR IF NO SECONDARY
1F1C    D15F    ERRQ 059                 063     GOINT      BR IF G1 BIT5=1    BR IF ACTIVE
1F1E    CA23    ERRQ 060                 062     INTFCK     BR IF HO BIT4=1    NOT ACTIVE, BR IF NO CHE
1F20    DA5F    ERRQ 061                 063     GOINT      BR IF HO BIT5=1    BR IF DE
1F22    AB20    ERRQ 062    INTFCK       BSWI 107  RESTRH    BR                 RETURN TO NATIVE REQUEST
1F5E    8D68    ERRQ 063    GOINT        DPTR 075  TAKEIN    BR                 GO FOR IB,STORE UCW, GO BSWI
1F06    2406    ERRQ 064    CPUXFR                          SET MODE K=30      RDR MODE, CPU ZONE
1F08    0B30    ERRQ 065                                    RST R K=03         RST RDR REQUEST & HWDE(RU015)
1F0A    9816    ERRQ 066                 CCOM 058  LSAVEB    BAL                SAVE P, I
1F0C    8A68    ERRQ 067                 091     RDIND      BAL                READ INDICATORS & SENSE
1F0E    999E    ERRQ 068                 ERDR 123  SSDONE     BR                 GO TO XFER BUFR
                ERRQ 070  *
                ERRQ 071  *                                    BRANCH AND LINK ROUTINES
                ERRQ 072  *
0F6C    4642    ERRQ 074    RRDUCW                          RDH D DC,98        READ CONSTANT OF 00C8
0F6E    3715    ERRQ 075                                    D1=D1$K10            D=00D8
0F70    F768    ERRQ 076    RDUCW            088  PHUCW     BR IF D13=0          BR IF PUNCH
0F72    2406    ERRQ 077                                    SET MODE K=30        SET READER MODE
0F74    3743    ERRQ 079    RDCONT                          D1=D1$K04          D=0XXC
0F76    5060    ERRQ 080                                    RDH U AS,D+0       READ COUNT INTO U
0F78    1773    ERRQ 081    RDSTCH                          D1=D1*-K07         D=0XX8
0F7A    5468    ERRQ 082    RDCHST                          RDH G AS,D+2        READ CHAN STAT FLAGS
0F7C    5E6A    ERRQ 083                                    RDH H AS,D-2        READ UNIT STAT
0F7E    128E    ERRQ 084                                    RTN
0F64    4642    ERRQ 086    RPHUCW                          RDH D DC,98        READ CONSTANT OF 00C8
0F66    3725    ERRQ 087                                    D1=D1$K20            D=00E8
0F68    2402    ERRQ 088    PHUCW                           SET MODE K=10        SET PUNCH MODE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS
0F6A    C4F4    ERRQ 089                 079     RDCONT     BR IF ZNZ
0A68    2A15    ERRQ 091    RDIND                           T0=0$K10
0A6A    F774    ERRQ 092                 097     RPHIND     BR IF D1 BIT3=0      BR IF PCH
0A6C    2406    ERRQ 093                                    SET MODE K=30        RDR MODE
0A6E    2BF5    ERRQ 094                                    T1=0$KF0             T=10F0
0A70    5CAA    ERRQ 095    READ                            RDH P AS,T-2         P0=INDICATORS, P1=SENSE
0A72    128E    ERRQ 096                                    RTN
0A74    2402    ERRQ 097    RPHIND                          SET MODE K=10        PCH MODE
0A76    2BB1    ERRQ 098                                    T1=0-K0B             T=10F4
0A78    C4F0    ERRQ 099                 095     READ       BR IF ZNZ            UNCONDITIONAL BR
33E8    2A15    ERRQ 101    STIND                           T0=0$K10
33EA    F772    ERRQ 102                 106     SPHIND     BR IF D1 BIT3=0      BR IF PH
33EC    2BF5    ERRQ 103                                    T1=0$KF0             T=10F0
33EE    7CAA    ERRQ 104    STORE                           STH P AS,T-2         STORE INDICATORS & SENSE
33F0    128E    ERRQ 105                                    RTN
33F2    2BB1    ERRQ 106    SPHIND                          T1=0-K0B             T=10F4
33F4    C4EE    ERRQ 107                 104     STORE      BR IF ZNZ            UNCONDITIONAL BR
343C    F744    ERRQ 109    GUESS        113     PCH2       BR IF D1 BIT3=0      BR IF PCH
343E    D243    ERRQ 110                 112     RETUN      BR IF D0 BIT5=1      BR IF FEED ONLY
3440    0D04    ERRQ 111                                    RST RP K=20          RST RDR CMD INLK(RU021)
3442    128E    ERRQ 112    RETUN                           RTN
3444    0D80    ERRQ 113    PCH2                            RST RP K=08          RST PCH CMD INLK(RU031)
3446    128E    ERRQ 114                                    RTN
                                        ************************************
                                        * CROSS REFERENCE FOR CSECT ERRQ   *
                                        ************************************
ERRQ 004    DCLA 394  DCLA 395
ERRQ 008    ERRQ 006
ERRQ 012    ERRQ 014
ERRQ 013    ERRQ 008
ERRQ 015    ERRQ 010  ERRQ 011
ERRQ 017    BSWI 070
ERRQ 018    ERRQ 013
ERRQ 030    BSWI 064
ERRQ 031    ERRQ 009
ERRQ 036    ERRQ 020
ERRQ 037    ERRQ 021  ERRQ 029  ERRQ 056
ERRQ 040    ERRQ 038
ERRQ 049    DCLR 038  ERDR 066  ERDR 144  ERRQ 044  ERRQ 045
ERRQ 050    ERRQ 041  ERRQ 055
ERRQ 054    ERRQ 052
ERRQ 055    ERRQ 018  ERRQ 034
ERRQ 062    ERRQ 058  ERRQ 060
ERRQ 063    ERRQ 042  ERRQ 043  ERRQ 059  ERRQ 061
ERRQ 064    ERRQ 032
ERRQ 074    ERRQ 030  EXFR 109
ERRQ 076    ERRQ 004
ERRQ 079    DPTQ 004  ERRQ 089
ERRQ 081    BWRP 146
ERRQ 082    DCLA 274  DCLR 030  DPTS 050  DPTT 007  DYPE 208  DYPE 221  ERDR 118
ERRQ 086    BWRP 024  ECOL 064  ERRQ 017  EXFR 092
```

CROSS REFERENCE FOR CSECT ERRQ

```
ERRQ 088     ERRQ 076
ERRQ 091     ECOL 065  ERDR 057  ERDR 068  ERRQ 067  EXFR 094  EXFR 110
ERRQ 095     ERRQ 099
ERRQ 097     EPRQ 092
ERRQ 101     EPCH 068  ERCX 049  ERDR 125  ERDR 140  ERRQ 025  EXFR 126
ERRQ 104     ERRQ 107
ERRQ 106     ERRQ 102
ERRQ 109     DCLR 052  ERDR 086  ERDR 102  ERDR 141
ERRQ 112     ERRQ 110
ERRQ 113     ERRQ 109
```

ETRP DESCRIPTIVE TEXT

ENTRY POINTS

(0110)	THIS IS THE ASSIGNED TRAP ADDRESS THAT IS FORCED INTO THE SAR WHEN A 2540 PUNCH TRAP IS ALLOWED. DIAGRAM 5-115 IN THE FEMDM SHOWS THE READER TRAP HANDLING.	INLK	ENTRY IS MADE HERE FROM THE BMCK ROUTINE. A MACHINE CHECK HAS BEEN DETECTED DURING A PUNCH TRAP, THE BMCK ROUTINE HAD BEEN TRAPPED TO, THE MMSK AND BA EXTERNALS HAVE BEEN SAVED AND THE LOGOUT LATCH HAS BEEN SET. THE OBJECT OF THIS ENTRY IS TO RESET COMMAND INTERLOCK AND RETURN TO THE MICROROUTINE TRAPPED FROM.

PROCEDURE FOR ANALYZING PUNCH CHECKS DUE TO HOLE COUNT

-1- SET ADDRESS SWITCHES TO ADDRESS OF LABEL -PCHCHK-

-2- SET MODE SWITCH TO SAR DELAYED STOP

-3- WHEN THE ERROR OCCURS, THE PUNCH ATTENTION LITE WILL COME ON AND THE FOLLOWING AREAS MAY BE DISPLAYED -

---PUNCH CHECK LOGOUT AREA---

AUX	50F6	50F7	50F8	50F9	50FA	50FB	50FC	50FD	50FE	50FF
BITS	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567
COLS	1 THRU 8	9 THRU 16	17 THRU 24	25 THRU 32	33 THRU 40	41 THRU 48	49 THRU 56	57 THRU 64	65 THRU 72	73 THRU 80

A BIT BEING ON IN THE LOGOUT AREA SIGNIFIES AN ERROR FOR THAT COLUMN.
THIS AREA MUST BE LOADED WITH BLANKS AFTER EVERY ERROR FOR SUBSEQUENT ERROR ANALYSIS.

---ROW FORM PUNCH IMAGE---

ROW	AUX ADDRESS	ROW	AUX ADDRESS
12	400C - 4015	11	4016 - 401F
0	4020 - 4029	1	402A - 4033
2	4034 - 403D	3	403E - 4047
4	4048 - 4051	5	4052 - 405B
6	405C - 4065	7	4066 - 406F
8	4070 - 4079	9	407A - 4083

A BIT BEING OFF IN THE ROW FORM PUNCH IMAGE BUFFER SIGNIFIES A HOLE IN THE CARD. THE CARD IS IN THE STACKER POKET.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		ETRP 001	T			NATIVE PUNCH TRAP ROUTINE	

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              ETRP 002    ATABLE  ADDR=0110
              ETRP 009    *
              ETRP 010    *
              ETRP 011    *               THE REGISTERS ARE USED FOR THE FOLLOWING
              ETRP 012    *
              ETRP 013    *                     G0-TRAP COUNT
              ETRP 014    *                     V -AUX STORAGE ADDRESS MODIFICATION
              ETRP 015    *                     D05=1 NORMAL WRITE INDICATOR
              ETRP 016    *                     D07=1 12 ROW PUNCH
              ETRP 017    *
              ETRP 018    *
              ETRP 019    *
0110    2208  ETRP 020                                 LINK U MMSK4=1
0112    A488  ETRP 021                      023  START  BR
              ETRP 022    AEND
2488    FADA  ETRP 023     START            072  SETUP  BR IF RPS BIT7=0      BR IF CLUTCH TRAP(SX044)
248A    5ACF  ETRP 024                                 P0=RP2               GET PUNCH CHECK BR INFO
248C    5BFF  ETRP 025                                 H1=RP1               GET PFR INFORMATION
248E    5F2C  ETRP 026                                 RDB P0 AS,V+1        SEND PUNCH DATA
2490    F245  ETRP 027                      057  ROW12  BR IF D0 BIT7=1      BR IF 12TH ROW
2492    D216  ETRP 029                      031  PFR    BR IF D0 BIT5=0      BR IF PFR OPERATION
2494    DFAB  ETRP 030                      041  NOTPFR BR IF PS BIT1=1      BR IF NOT NORMAL RUN IN(SX041)
2496    236D  ETRP 031    PFR                          V1=V1+K60            POINT TO
2498    23C8  ETRP 032                                 V1=V1+K0C              PFR AREA
249A    5D20  ETRP 033                                 RDB P1 AS,V+0        READ PFR ROW AREA
249C    7F20  ETRP 034                                 STB H1 AS,V+0        STORE NEW PFR INFO
249E    337F  ETRP 035                                 V1=V1-K77            POINT TO PCH IMAGE
24A0    5F20  ETRP 036                                 RDB H1 AS,V+0
24A2    1FFF  ETRP 037                                 H1=H1<KFF
24A4    6DF1  ETRP 038                                 P1=P1<H1
24A6    7D2C  ETRP 039                                 STB P1 AS,V+0        STORE RESULTS
24A8    23C8  ETRP 040                                 V1=V1+K0C
24AA    223D  ETRP 041    NOTPFR                       V0=V0+K30            POINT ADDRESS TO CHECK AREA
24AC    5F20  ETRP 045                                 RDB H1 AS,V+0        READ CHECK IMAGE BYTE
24AE    6CF1  ETRP 046                                 P0=P0<H1             COMPARE RESULTS
24B0    C4C3  ETRP 047                      056  NOEROR BR IF Z=0            BR IF NO ERROR
24B2    CAC2  ETRP 048                      056  NOEROR BR IF RPS BIT4=0     BR IF NO PCH CHK CL (SX044)
24B4    3F40  ETRP 049    PCHCHK                       SET P K=84           ERR,SET PCH CHK $ RESTART(RU031)
24B6    2E55  ETRP 050                                 H0=0$K50             SET ERROR LOG OUT
24B8    54FD  ETRP 051                                 H1=G0L                ADDRESS TO
24BA    3FF5  ETRP 052                                 H1=H1$KF0               50F6 - 50FF
24BC    5DE0  ETRP 053                                 RDB P1 AS,H+0        READ OLD ERROR BYTE
24BE    6DC5  ETRP 054                                 P1=P1$P0             OR NEW ERROR TO OLD BYTE
24C0    7DE0  ETRP 055                                 STB P1 AS,H+0        STORE NEW RESULTS
24C2    2215  ETRP 056    NOEROR                       V0=0$K10             POINT BACK TO PCH IMAGE AREA
24C4    241B  ETRP 057    ROW12                        G0=G0+K01            UP DATE COUNT
24C6    F0D0  ETRP 058                      063  RETURN BR IF LZNZ           BR IF NOT THE LAST TRAP OF A ROW
24C8    F24C  ETRP 059                      061  G0ADD6 BR IF D0 BIT7=0      BR IF NOT 12TH ROW
24CA    1613  ETRP 060                                 D0=D0*-K01           RST 12TH ROW INDICATOR
24CC    246B  ETRP 061    G0ADD6                       G0=G0+K06            CORRECT G0 TO X6
24CE    E0D3  ETRP 062                      065  DONE   BR IF HZ=0           BR IF COUNT FINISHED
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS

24D0    0208    ETRP 063    RETURN                       RTN U MMSK4=0
24D2    DF80    ETRP 065    DONE          079  RUNIN     BR IF PS1=0          FINISHED BR IF NORM RUNIN(SX041)
24D4    CFD8    ETRP 069                  071  INLK      BR IF PS BIT0=0      BR IFNOT READY(SX041)
24D6    2D10    ETRP 070                               SET RP K=01          SET PCH DE (RU016)
24D8    0D80    ETRP 071    INLK                       RST RP K=C8          RST PCH CMD INLK(RU031)
24DA    2215    ETRP 072    SETUP                      V0=0$K10
24DC    2307    ETRP 073                               V1=0                 V=1000
24DE    5F2C    ETRP 074                               RDB P0 AS,V+1        PRE SHIP PUNCH DATA
24E0    3613    ETRP 075                               D0=D0$K01            SET 12TH ROW INDICATOR
24E2    2435    ETRP 076                               G0=0$K30             SET COUNT TO 36 FOR 130
24E4    A4CC    ETRP 077                  061  GOADD6    BR                    TRAPS IN DECIMAL
2480    23B1    ETRP 079    RUNIN                      V1=0-K0B
2482    2DC7    ETRP 080                               P1=0
2484    7D20    ETRP 081                               STB P1 AS,V+0
2486    A4D8    ETRP 082                  071  INLK      BR
                                          ************************************
                                          * CROSS REFERENCE FOR CSECT ETRP *
                                          ************************************

ETRP 023    ETRP 021
ETRP 031    ETRP 029
ETRP 041    ETRP 030
ETRP 056    ETRP 047  ETRP 048
ETRP 057    ETRP 027
ETRP 061    ETRP 059  ETRP 077
ETRP 063    ETRP 058
ETRP 065    ETRP 062
ETRP 071    BMCK 074  ETRP 069  ETRP 082
ETRP 072    ETRP 023
ETRP 079    ETRP 065
```

ETTR DESCRIPTIVE TEXT

ENTRY POINTS

(01B0)	THIS IS THE ASSIGNED TRAP ADDRESS THAT IS FORCED INTO THE SAR WHEN A 2540 READ TRAP IS ALLOWED. DIAGRAM 5-106 IN THE FEMDM SHOWS THE READER TRAP HANDLING.	RQUEST	ENTRY IS MADE TO THIS LABEL FROM THE BMCK ROUTINE. A MACHINE CHECK HAS BEEN DETECTED DURING A READER TRAP, THE BMCK ROUTINE HAD BEEN TRAPPED TO, THE MMSK AND BA EXTERNALS HAVE BEEN SAVED AND THE LOGOUT LATCH HAS BEEN SET. THE OBJECT OF THIS ENTRY IS TO SET A READER STATUS REQ AND RETURN TO THE ROUTINE TRAPPED FROM.

PROCEDURE FOR ANALYZING READER CHECKS DUE TO HOLE COUNT

-1- SET ADDRESS SWITCHES TO ADDRESS OF LABEL -RDRCHK-

-2- SET MODE SWITCH TO SAR DELAYED STOP

-3- WHEN THE ERROR OCCURS, THE RDR ATTENTION LITE WILL COME ON AND THE FOLLOWING AREAS MAY BE DISPLAYED -

--READER CHECK LOGOUT AREA---

AUX	10F6	10F7	10F8	10F9	10FA	10FB	10FC	10FD	10FE	10FF
BITS	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567	01234567
COLS	1 THRU 8	9 THRU 16	17 THRU 24	25 THRU 32	33 THRU 40	41 THRU 48	49 THRU 56	57 THRU 64	65 THRU 72	73 THRU 80

A BIT BEING ON IN THE LOGOUT AREA SIGNIFIES AN ERROR FOR THAT COLUMN.
THIS AREA MUST BE LOADED WITH BLANKS AFTER EVERY ERROR FOR SUBSEQUENT ERROR ANALYSIS.

---COLUMN FORM AREA--- AUX LOCATIONS 5018 - 50B7

A BIT BEING ON INDICATES A HOLE IN THE CARD THAT IS NOW IN THE PRE STACKER STATION. THIS IS A RESULT OF THE FIRST READ STATION TRAPS.

FORMAT	5018			5019			
	BIT	ROW	COL		ROW	COL	
	0	X		0	X		FORMAT IS THE
	1	X		1	X		SAME FOR EACH
	2	12		2	4		CARD COLUMN.
	3	11	-1-	3	5	-1-	AUX 501A-501B
	4	0		4	6		FOR COL 2, AUX
	5	1		5	7		501C-501D FOR
	6	2		6	8		COL 3, ETC.
	7	3		7	9		

---ROW FORM AREA--- AUX LOCATIONS 4FA0 - 5017

A BIT BEING A ZERO INDICATES A HOLE IN THE CARD THAT IS BETWEEN THE FIRST AND SECOND READ STATIONS.

ROW	AUX ADDRESS	ROW	AUX ADDRESS
9	4FA0 - 4FA9	3	4FDC - 4FE5
8	4FAA - 4FB3	2	4FE6 - 4FEF
7	4FB4 - 4FBD	1	4FF0 - 4FF9
6	4FBE - 4FC7	0	4FFA - 5003
5	4FC8 - 4FD1	11	5004 - 500
4	4FD2 - 4FD8	12	500E - 5017

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

                ETTR 001    T              NATIVE READER TRAP ROUTINE
                ETTR 002    ATABLE  ADDR=01B0
                ETTR 009    *
                ETTR 010    *
                ETTR 011    *
                ETTR 012    *              THE REGISTERS IN  THIS ROUTINE ARE USED AS FOLLOWS
                ETTR 013    *
                ETTR 014    *                        D0 BIT2=1  XFER ROW TO COL
                ETTR 015    *                        D1 REG-TRAP COUNT
                ETTR 016    *                        U  REG-ROW ADDRESS
                ETTR 017    *
01B0    2206    ETTR 018                                LINK U MMSK3=1
01B2    5ACF    ETTR 019                                P0=RP2                 GET READ 2 INFO
01B4    B204    ETTR 020                  022  START    BR
                ETTR 021    AEND
3204    EAB4    ETTR 022     START        048  SETUP    BR IF RPS BIT6=0       BR IF SET-UP(SX044)
3206    58DF    ETTR 023                                P1=RP1                 GET READ 1 INFO(START SHIFT REG)
3208    5F00    ETTP 024                                RDB H1 AS,U+0          READ OLD DATA
320A    7D08    ETTR 025                                STB P1 AS,U+1          STORE NEW DATA
320C    6CF1    ETTR 026                                P0=P0<H1               CHECK OLD READ 1 WITH NEW READ 2
320E    C4A1    ETTR 027                  036  NOEROR   BR IF Z=0              BR IF NO ERROR
3210    CDA1    ETTR 028                  036  NOEROR   BR IF RS BIT0=1        BR IF NOT GATE RD CMPLTE(SX041)
3212    3B00    ETTR 029     RDRCHK                     SET R K=80             SET ERROR LAT (RU021)
3214    2E15    ETTR 030                                H0=0$K10               SET H TO ADDRESS READ CHECK
3216    57ED    ETTR 031                                H1=D1L                   LOG OUT AREA IN AUX STORAGE
3218    2FFD    ETTR 032                                H1=H1+KF0                POSITION 10F6 THRU 10FF
321A    5DE0    ETTR 033                                RDB P1 AS,H+0          READ OLD ERROR BYTE
321C    6DC5    ETTR 034                                P1=P1$P0               OR NEW ERROR
321E    7DE0    ETTR 035                                STB P1 AS,H+0          STORE ORED RESULTS
3220    271B    ETTR 036    NOEROR                      D1=D1+K01              UPDATE TRAP COUNT
3222    F0A8    ETTR 037                  040  RETURN   BR IF LZNZ
3224    276B    ETTR 038    D1ADD6                      D1=D1+K06              CORRECT D1 LOW TO X6
3226    E0AB    ETTR 039                  041  DONE     BR IF HZ=0             BR IF FINISHED
3228    0206    ETTR 040    RETURN                      RTN U MMSK3=0
322A    CD81    ETTR 041    DONE          046  XFER     BR IF RS BIT0=1        BR IF NOT GATE RD CMPLTE(SX041)
322C    DDB0    ETTR 042                  044  INLK     BR IF RS BIT1=0        BR IF NOT READY(SX041)
322E    2D20    ETTR 043    RQUEST                      SET RP K=02            SET RDR DE (RU016)
3230    0D04    ETTR 044    INLK                        RST RP K=20            RST CMD INLK (RU021)
3232    B236    ETTR 045                  049  SURE     BR                     MAKE SURE ADDR SITTING RIGHT
3200    3625    ETTR 046    XFER                        D0=D0$K20              SET XFER BIT
3202    B22E    ETTR 047                  043  RQUEST   BR
3234    58DF    ETTR 048    SETUP                       P1=RP1                 GET READ 1 INFO(START SHIFT REG)
3236    30B9    ETTR 049    SURE                        U0=0-KB0
3238    21A5    ETTR 050                                U1=0$KA0               U=4FA0
323A    2745    ETTR 051                                D1=0$K40
323C    B224    ETTR 052                  038  D1ADD6   BR                     GO MAKE D1=46
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT ETTR *
                                  **********************************

ETTR 022     ETTR 020
ETTR 036     ETTR 027  ETTR 028
ETTR 038     ETTR 052
ETTR 040     ETTR 037
ETTR 041     ETTR 039
ETTR 043     BMCK 072  ETTR 047
ETTR 044     ETTR 042
ETTR 046     ETTR 041
ETTR 048     ETTR 022
ETTR 049     ETTR 045
```

EXFR DESCRIPTIVE TEXT

ENTRY POINTS

XFER — ENTRY IS MADE HERE FROM THE ERDR ROUTINE TO SET UP THE ADDRESSES OF THE COLUMN AND ROW IMAGE BUFFERS, AND START THE TRANSFER OF ROW IMAGE TO COLUMN BINARY FORM.

BYTECT — ENTRY IS MADE HERE FROM THE ERDR ROUTINE AFTER THE ADDRESSES FOR PFR ROW IMAGE AND PFR COLUMN IMAGE BUFFERS HAVE BEEN SET UP. THE TRANSFER OF PFR ROW IMAGE TO PFR COLUMN IMAGE BEGINS HERE.

CHKCHK — ENTRY IS MADE HERE FROM THE ERDR ROUTINE WHEN A READ-NO-FEED OPERATION IS DETECTED. A CHECK IS MADE FOR READ AND VALIDITY ERRORS. IF NO ERRORS DETECTED, GO TO STORE INDICATOR AND SENSE BYTES IN AUXILIARY STORAGE.

COMBIN — ENTRY IS FROM THE ERCX ROUTINE WHEN A VALIDITY CHECK IS DETECTED DURING A PFR OPERATION. UNIT CHECK BIT IS SET, COMMAND CHAINING IS BROKEN, AND THE INDICATOR AND SENSE BYTES ARE STORED IN AUXILIARY STORAGE.

WLRBR — ENTRY IS FROM THE ECOL ROUTINE. THE WLR INDICATOR BIT HAD BEEN SET IF AN INCORRECT COUNT WAS DETECTED IN THE ECOL ROUTINE. THE ENTRY HERE IS TO TEST FOR SLI AND CHAINING, PRIOR TO STORING THE INDICATOR AND SENSE BYTES IN AUXILIARY STORAGE.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		EXFR 001	T			READER XFER OF ROW IMAGE TO COLUMN BINARY FORM	
		EXFR 005	ASEQ	ALO7=50			
1150	2455	EXFR 007	XFER			G0=0$K50	SET G TO 2ND BYTE OF
1152	2515	EXFR 008				G1=0$K10	COL BUFFER
1154	3593	EXFR 009				G1=G1$K09	ADDR 5019
1156	2045	EXFR 010				U0=0$K40	SET U TO ROW IMAGE ADDRESS
1158	21A5	EXFR 011				U1=0$KA0	40A0
115A	22A3	EXFR 012	BYTECT			V0=0$K0A	10 GROUPS OF 8 BYTES CNTR
115C	2363	EXFR 013	BITCT			V1=0$K06	6 BIT PER BYTE CNTR
115E	4852	EXFR 014				RDH I DC,9A	I=0000
1160	4A86	EXFR 015				T=I	CLEAR
1162	4C86	EXFR 016				P=I	WORK
1164	4E86	EXFR 017				H=I	REGISTERS
1166	2613	EXFR 018				D0=0$K01	SET MASK FOR COL 1
1168	5700	EXFR 019	RDBYTE			RDB D1 AS,U	READ BYTE FROM IMAGE
116A	C76F	EXFR 020		C22	N	BR IF D1 BIT0=1	BITS ARE COMPLIMENT
116C	6965	EXFR 021				I1=I1$D0	OR BIT IN COL 1
116E	D773	EXFR 022	N	024	NE	BR IF D1 BIT1=1	
1170	6865	EXFR 023				T1=T1$D0	COL 2
1172	E777	EXFR 024	NE	026	NEX	BR IF D1 BIT2=1	
1174	6D65	EXFR 025				P1=P1$D0	COL 3
1176	F77B	EXFR 026	NEX	028	NEXT	BR IF D1 BIT3=1	
1178	6F65	EXFR 027				H1=H1$D0	COL 4
117A	C37F	EXFR 028	NEXT	C30	NEXTB	BR IF D1 BIT4=1	
117C	6865	EXFR 029				I0=IC$D0	COL 5
117E	D303	EXFR 030	NEXTB	032	NEXTBT	BR IF D1 BIT5=1	
1180	6A65	EXFR 031				T0=T0$D0	COL 6
1182	E307	EXFR 032	NEXTBT	034	BTNEXT	BR IF D1 BIT6=1	

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

1184  6C65    EXER 033                                    P0=P0$D0               COL 7
1186  F30B    EXER 034    BTNEXT        036   BTDONE     BR IF D1 BIT7=1
1188  6E65    EXER 035                                    H0=H0$D0               COL 8
118A  6663    EXER 036    BTDONE                          D0=D0+D0               SHIFT MASK LEFT ONE
118C  23FF    EXER 037                                    V1=V1+KFF              BIT COUNT MINUS ONE
118E  C49F    EXER 038                  050   STORE      BR IF Z=0
1190  F41A    EXER 040                  047   RPFR1      BR IF GO BIT3=0        BR IF PFR READ
1192  21AB    EXER 042                                    U1=U1+K0A              NO- INCREMENT ADR
1194  F498    EXER 043                  045   READM      BR IF AC=0
1196  201D    EXER 044                                    U0=U0+K10              UPDATE TO 60
1198  9168    EXER 045    READM         019   RDBYTE     BR
119A  319B    EXER 047    RPFR1                           U1=U1-K09              DECREMENT ADR BY 10
119C  9168    EXER 048                  019   RDBYTE     BR
119E  7940    EXER 050    STORE                           STB I1 AS,G+0          STORE COL ONE OR FIVE
11A0  6444    EXER 051                                    G=G+2      **POINT ADDR TO EVERY OTHER BYTE
11A2  7B40    EXER 052                                    STB T1 AS,G+0          COL TWO OR SIX
11A4  6444    EXER 053                                    G=G+2
11A6  7D40    EXER 054                                    STB P1 AS,G+0          COL THREE OR SEVEN
11A8  6444    EXER 055                                    G=G+2
11AA  7F40    EXER 056                                    STB H1 AS,G+0          COL FOUR OR EIGHT
11AC  6444    EXER 057                                    G=G+2
11AE  144B    EXER 058                                    G0=G0<K04      **INVERT BIT 5 - 2ND 4 BYTES
11B0  D03C    EXER 059                  065   STRDON     BR IF GO BIT5=0        BR IF DONE 2ND 4 BYTE
11B2  5899    EXER 060                                    I1=I0                  MOVE
11B4  5AB9    EXER 061                                    T1=T0                    2ND
11B6  5CD9    EXER 062                                    P1=P0                    4
11B8  5EF9    EXER 063                                    H1=H0                    BYTES
11BA  919E    EXER 064                  050   STORE      BR
11BC  22FF    EXER 065    STRDON                          V0=V0+KFF              GROUP CNTR MINUS
11BE  C4CF    EXER 066                  078   HALF       BR IF Z=0              BR IF DONE HALF
11C0  F44A    EXER 068                  075   RPFR2      BR IF GO BIT3=0        BR IF PFR READ
11C2  313D    EXER 070                                    U1=U1-K30              NEXT GROUP OF 8 BYTES
11C4  C048    EXER 071                  073   HAF1ST     BR IF GO BIT4=0        BR IF 1ST HALF
11C6  20FD    EXER 072                                    U0=U0+KF0              2ND HALF, 50XX
11C8  915C    EXER 073    HAF1ST        013   BITCT      BR                     GO DO ANOTHER GROUP
11CA  213F    EXER 075    RPFR2                           U1=U1+K33              NEXT GROUP OF 8 BYTES
11CC  915C    EXER 076                  013   BITCT      BR                     GO DO ANOTHER GROUP
11CE  148B    EXER 078    HALF                            G0=G0<K08      **INVERT BIT 4 - 2ND HALF
11D0  C060    EXER 079                  090   DONE       BR IF GO BIT4=0        BR IF DONE 2ND HALF
11D2  F45A    EXER 081                  087   RPFR3      BR IF GO BIT3=0        BR IF PFR READO
11D4  211B    EXER 083                                    U1=U1+K01              U=40DC
11D6  35AD    EXER 084                                    G1=G1-KA0              G=5918
11D8  915A    EXER 085                  012   BYTECT     BR                     GO DO 2ND HALF
11DA  21A7    EXER 087    RPFR3                           U1=0$KAA               U=10AA  PFR READ
11DC  2507    EXER 088                                    G1=0                   G=4900
11DE  915A    EXER 089                  012   BYTECT     BR                     GO DO 2ND HALF
11E0  F475    EXER 090    DONE          100   ENDING     BR IF GO BIT3=1        BR IF NOT PFR READ
11E2  52E2    EXER 091                                    RDH V DA,BC            PFR READ,RESTORE DATA ADDRESS
11E4  8F64    EXER 092              ERRQ 086  RPHUCW     BAL                    SET D1 TO E8
11E6  51A2    EXER 093                                    RDB U1 DA,AC           RESTORE CCW ADJUSTED COUNT
11E8  8A68    EXER 094              ERRQ 091  RDIND      BAL                    P0=IND,P1=SENSE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
11EA    54B2    EXER 095                                    RDH G DA,AE        RESTORE CMD AND FLAGS
11EC    3659    EXER 096                                    D0=0-K50           WLR COUNT D0=AF
11EE    2845    EXER 097                                    I0=0$K40
11F0    2907    EXER 098                                    I1=0               I=4000 POINT TO PFR COL BUFFER
11F2    99DE    EXER 099            ERDR 164  BRSKIP        BR
11F4    2466    EXER 100  ENDING                            SET MODE K=36      READER MODE AND ZONE
11F6    E67E    EXER 104                 108  RUNON         BR IF DC BIT2=0    BR IF NO EXFR
11F8    1625    EXER 105                                    D0=D0*-K20         XFER,RST BIT
11FA    2400    EXER 106                                    SET MODE K=00      CPU MODE AND ZONE
11FC    AB20    EXER 107            BSWI 107  RESTRH        BR
11FE    2406    EXER 108  RUNON                             SET MODE K=30      READER MODE AND ZONE
1200    8F6C    EXER 109            ERRQ 074  RRDUCW        BAL                SET D=00D8
1202    8A68    EXER 110            ERRQ 091  RDIND         BAL
1204    5FB2    EXER 111                                    RDB H1 DA,AE       RESTORE CMD
1206    FB0A    EXER 112                 114  CHKCHK        BR IF H1 BIT7=0    BR IF NOT CMD IMMEDIATE
1208    2643    EXER 113                                    D0=0$K04           CMD IMEDIATE,SET INDICATOR
120A    FD8E    EXER 114  CHKCHK         116  NORDCK        BR IF RS BIT3=0    BR IF NO READ CHECK(SX042)
120C    3D15    EXER 115                                    P1=P1$K10          READ CHK,
120E    C992    EXER 116  NORDCK         118  COMBIN        BR IF RS BIT4=0    BR IF NO VAL CHK(SX043)
1210    3D83    EXER 117                                    P1=P1$K08          VAL CHK,SET INDICATOR IN SENSE
1212    2E23    EXER 118  COMBIN                            H0=0$K02           UNIT CHECK
1214    3485    EXER 119                                    G0=G0$K80          SET SECONDARY BIT
1216    A925    EXER 120                 125  BRKCC   N     BR IF P1=NZ        BR IF NO SENSE BIT
1218    2E07    EXER 121                                    H0=0               NO UNIT CHK
121A    F822    EXER 122  WLRBR          126  CCBR          BR IF P0 BIT7=0    BR IF NO WLR
121C    E523    EXER 123                 126  CCBR          BR IF G1 BIT2=1    WLR, BR IF SLI
121E    3445    EXER 124                                    G0=G0$K40          WLR
1220    1545    EXER 125  BRKCC  0                          G1=G1*-K40         RESET CC  BIT
1222    B3E8    EXER 126  CCBR           ERRQ 101  STIND     BAL                STORE IND., AND SENSE
1224    D529    EXER 127                 129  CC            BR IF G1 BIT1=1    BR IF CC
1226    3E83    EXER 128                                    H0=H0$K08          NO CC,SET CHAN END BIT
1228    D22E    EXER 129  CC             132  NOTIME        BR IF D0 BIT5=0    BR IF NOT CMD IMMEDIATE
122A    56F2    EXER 130                                    RDH D DA,BE        CMD IMMEDIATE, RESTORE D0 BIT 7
122C    8D6E    EXER 131            DPTR 082  IMMEDI        BR
122E    51A2    EXER 132  NOTIME                            RDB U1 DA,AC       RESTORE RESIDUE COUNT
1230    2007    EXER 133                                    U0=0               HIGH COUNT EQUALS ZERO
1232    D83A    EXER 134                 138  HICTO         BR IF P0 BIT5=0    BR IF NO EXCESS COUNT STORED
1234    52F2    EXER 135                                    RDH V DA,BE        READ EXCESS COUNT
1236    613B    EXER 136                                    U1C=U1+V1          ADD BACK THE
1238    602D    EXER 137                                    U0C=U0+V0+C         EXCESS COUNT
123A    C844    EXER 138  HICTO          143  NOIPL         BR IF P0 BIT4=0    BR IF FEED
123C    3E43    EXER 139                                    H0=H0$K04          NO FEED, SET DE
123E    C3C4    EXER 140                 143  NOIPL         BR IF BA BIT4=0    BR IF NOT IPL
1240    2B20    EXER 141                                    SET R K=02         IPL,SET RDR REQUEST(RU015)
1242    8D6A    EXER 142  GOSTUC    DPTR 076  STINST        BR                 STORE UCW & GO TO BSWI
1244    C3C3    EXER 143  NOIPL          142  GOSTUC        BR IF BA BIT4=1    NO INTERRUPT FOR IPL
1246    8D5A    EXER 144            DPTR 068  CCBR          BR                 INT, OR GO CC,STORE UCW, GO BSWI
                EXER 145  AEND
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT EXFR *
                                  **********************************
EXFR 007     ERDR 181  ERDR 183  ERDR 185
EXFR 012     ERDR 153  EXFR 085  EXFR 089
EXFR 013     EXFR 073  EXFR 076
EXFR 019     EXFR 045  EXFR 048
EXFR 022     EXFR 020
EXFR 024     EXFR 022
EXFR 026     EXFR 024
EXFR 028     EXFR 026
EXFR 030     EXFR 028
EXFR 032     EXFR 030
EXFR 034     EXFR 032
EXFR 036     EXFR 034
EXFR 045     EXFR 043
EXFR 047     EXFR 040
EXFR 050     EXFR 038  EXFR 064
EXFR 065     EXFR 059
EXFR 073     EXFR 071
EXFR 075     EXFR 068
EXFR 078     EXFR 066
EXFR 087     EXFR 081
EXFR 090     EXFR 079
EXFR 100     EXFR 090
EXFR 108     EXFR 104
EXFR 114     ERDR 119  EXFR 112
EXFR 116     EXFR 114
EXFR 118     ERCX 048  EXFR 116
EXFR 122     ECOL 094
EXFR 125     EXFR 120
EXFR 126     EXFR 122  EXFR 123
EXFR 129     EXFR 127
EXFR 132     EXFR 129
EXFR 138     EXFR 134
EXFR 142     EXFR 143
EXFR 143     EXFR 138  EXFR 140
```

FILE DESCRIPTIVE TEXT

2311 UCW

AUX 0X8E-0X8F	ZONE 1 T-REG	ZONE 1 I-REG	AUX 0X98-0X99
STATUS-I/O ADR	FLAGS AND OP (FFI) (FOP)	DATA ADDRESS	COUNT FIELD / NEXT CCW ADDR

STATUS-I/O ADR

0=ACTIVE
1=CAME FROM START I/O
2=RETURN TO ICYCLES
3=PERFORMED HALT I/O
4-15=FILE I/O ADDRESS

FLAGS AND OP

0=CHAIN DATA
1=CC OR CD AND NOT UNIT CHECK
2=CC OR CD AND NOT UNIT CHECK
3=SKIP
4=PCI
5=SET INTERRUPT LATCH (STATUS IN INTERRUPT BUFFER).
6=ANY INTERRUPT CONDTION
7=SLI

2311 AUX MODULE 0 USAGE

ADDRESS

C4 AND C5 - DEVICE ADDRESS FOR CONTINGENT CONNECTION

06 AND 07 - INTERRUPT BUFFER

14 AND 15 - ERROR STATUS (CE)

16 AND 17 - ERROR NEXT CCW ADDR

8E AND 8F - STATUS AND CHANNEL DEVICE ADDRESS.

98 AND 99 - NEXT CCW ADDRESS

9A AND 9B - SENSE BYTES 0 AND 1 OR PREVIOUS OP AND MASK.

9C AND 9D - SENSE BYTES 0 AND 1 OR FILE ADDRESS

F5 - CONTROL UNIT ADDRESS

ENTRY POINTS

ENTRY — THIS IS THE NORMAL ENTRY FOR A PROGRAMMED INSTRUCTION. THIS ROUTINE DETERMINES- START I/O, TEST I/O, TEST CHANNEL, OR HALT I/O AND BRANCHES FOR REQUIRED PROCESS. THE START I/O SEQUENSE TESTS THE AVAIL-ABILITY OF THE FILE, AND DETERMINES THE FILE READY CONDITION. THE FILE CONTROLS ARE TESTED FOR OPERATION.

TSTCC — THIS ENTRY ALLOWS THE USE OF THE CYCLIC-CODE AND OPERATIONAL TEST ROUTINE AFTER CONTROL AND SENSE OPERATIONS BEFORE PROCEEDING WITH OPERATING COMMANDS.

SNSSTA — THIS IS A BAL ENTRY TO SET AN EQUIPMENT CHECK AND SELECTED STATUS INDICATIONS IN THE SENSE BYTES.

SETDA — THIS IS A BAL ENTRY USED TO SET THE DEVICE ADDRESS FROM AN UNSELECTED GATED ATTENTION. AFTER DEVELOPING THE DEVICE ADDRESS, THE INDICATED MODULE IS SELECTED.

RSTATT — THIS IS A BAL ENTRY USED TO RESET THE FILE GATED ATTENTION. THE READ-GATE IS RAISED MOMENTARILY FOR THE RESET AND THEN THE MODULE IS DESELECTED.

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		FILE 001	T			2311 FEATURE COMMON ENTRY FOR START IO,TEST IO,HALT IO	
		FILE 002	*			**********	
		FILE 003	*			*	**
		FILE 004	*			FILE TIO - SIO - HIO COMMON	**
		FILE 005	*			*	**

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

              FILE 006    *          ************************************************************************
              FILE 007    ASEQ    AL07=7C
047C    7CC2  FILE 008     ENTRY                             STH P DA,B8            STORE CPU P REG
047E    2480  FILE 009                                       SET MODE K=08          SET FILE MODE & CPU ZONE
0480    C62A  FILE 010                  031   AVAIL         BR IF D0 BIT0=0        BR IF CHANNEL AVAILABLE
0482    E013  FILE 011                  019   HALTIO        BR IF G0 BIT6=1        BR IF HALT IO
0484    F020  FILE 012                  026   SETCC2        BR IF G0 BIT7=0        BR IF NOT TEST IO
0486    D9A0  FILE 013                  026   SETCC2        BR IF FFI5=0           BR IF INTURRUPT LATCH NOT ON
0488    6711  FILE 014                                       D1=D1<U1               COMPARE IOA
048A    C4A0  FILE 015                  026   SETCC2        BR IF ZNZ              BR IF IOA NOT SAME
048C    7032  FILE 016                                       STH U DA,8E            SAVE IOA WITH RESET STATS
048E    9130  FILE 017              FINT 050   SETCSW        BAL                    SET REGS TO USE CPU CSW STORE
0490    AC24  FILE 018              CCOM 127   CSWADR        BR
0492    5DBF  FILE 019     HALTIO                           T1=FF0                 GET CCW FLAGS
0494    1BC5  FILE 020                                       T1=T1*-KC0             STRIP CC AND DC FLAGS
0496    D9A3  FILE 021                  027   SETHIO        BR IF FFI5=1           BR IF INTURRUPT LATCH IS ON
0498    3210  FILE 022                                       SET MMSK K=81          SET  PRIORITY
049A    2D04  FILE 023                                       SET FIA K=20           SET CCW COUNT 000
049C    4DBF  FILE 024                                       FF0=T1                 RESTORE CCw FLAGS WITHOUT CHAIN
049E    1210  FILE 025                                       RST MMSK K=81          RESET PRIORITY
04A0    AECA  FILE 026     SETCC2       CCOM 162   CC2B      BR                     BR TO SET CC2
04A2    F5BD  FILE 027     SETHIO            040   SETCUB    BR IF DASI3=1          BR IFERASING
04A4    2E07  FILE 028                                       H0=0                   ZERO DEVICE STATUS
04A6    D9C0  FILE 029                  042   HTS           BR IF FFI5=0           BR IF NOT INTURRUPT LATCH
04A8    810C  FILE 030              CCOM 155   CC0B          BR                     BR TO SET CC0
04AA    3065  FILE 031     AVAIL                            U0=U0$K60              SET FROM SIO&RTN TO I CYC STATS
04AC    7032  FILE 032                                       STH U DA,8E            STORE STATS AND I/O ADDRESS
04AE    E023  FILE 033                  027   SETHIO        BR IF G0 BIT6=1        BR IF HALT I/O
04B0    F5BD  FILE 034                  040   SETCUB        BR IF DASI3=1          BR IF ERASING
04B2    E1C3  FILE 035                  043   TSTCUE        BR IF DASI6=1          BR IF CUE
04B4    F056  FILE 036                  053   SIO           BR IF G0 BIT7=0        BR IF SIO
04B6    DBFD  FILE 037                  076   GETDS         BR IF FOP5=1           BR IF CONTIN CONNECT CLEAR
04B8    8224  FILE 038                  258   TSTADR        BAL                    TEST FOR CONTINGENT CONNECTION
04BA    DBFC  FILE 039                  076   GETDS         BR IF FOP5=0           BR IF NOT CONTIN CONNECT
04BC    2E55  FILE 040     SETCUB                           H0=0$K50               SET CUB STATUS
04BE    2D02  FILE 041                                       SET FIA K=10           SET CUB LATCH
04C0    855E  FILE 042     HTS          125   SIOTIO        BR
04C2    DBCC  FILE 043     TSTCUE       048   CUE           BR IF FOP5=0           BR IF NO CONTIN CONNECT
04C4    8224  FILE 044                  258   TSTADR        BAL                    TST FOR CONTIN CONNECT
04C6    DBBD  FILE 045                  040   SETCUB        BR IF FOP5=1           BR IF CONTIN CONNECT
04C8    2E43  FILE 046                                       H0=0$K04                     SET CONTINGENT
04CA    4FEF  FILE 047                                       FOP=H0                  CONNECTION CLEAR INDICATOR
04CC    2E25  FILE 048     CUE                              H0=0$K20               SET CUE FOR STATUS
04CE    F053  FILE 049                  051   RSTCUE        BR IF G0 BIT7=1        BR IF TEST I/O
04D0    2E35  FILE 050                                       H0=0$K30                  SET CUE & BUSY STATUS
04D2    0D02  FILE 051     RSTCUE                           RST FIA K=10           RESET CUB LATCH
04D4    855E  FILE 052                  125   SIOTIO        BR
04D6    7242  FILE 053     SIO                              STH V DA,98            SAVE CCW ADDRESS
04D8    42EF  FILE 054                                       STP0=H0                SET STORAGE PROTECT KEY
04DA    5A30  FILE 055                                       RDH T V+0              GET CCW COMMAND
04DC    0A4B  FILE 056                                       Z=T0<K04               MASK FOR SENSE OP
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
04DE  C4E4  FILE 057                 060     NOTSNS     BR IF ZNZ          BR IF NOT SENSE
04E0  8FA6  FILE 058                 316     MODSEL     BAL                SELECT MODULE FOR SENSE
04E2  910C  FILE 059                 236     EXCUTE     BR
04E4  DBEB  FILE 060  NOTSNS         063     TSTNOP     BR IF FOP5=1       BR IF CONTIN CONECT CLEAR
04E6  8224  FILE 061                 258     TSTADR     BAL                TST FOR CONTINGENT CONNECTION
04E8  DBBD  FILE 062                 040     SETCUB     BR IF FOP5=1       BR IF CONTIN CONECT
04EA  0A3B  FILE 063  TSTNOP                            Z=TO<K03           MASK FOR NO OP
04EC  C4FD  FILE 064                 076     GETDS      BR IF Z=0          BR IF NO OP
04EE  0A1D  FILE 065                                    Z=TO<K10           MASK FOR HI BIT OF RESTORE
04F0  E0F6  FILE 066                 070     TSTIOA     BR IF HZNZ         BR IF NOT RESTORE
04F2  0A7B  FILE 067                                    Z=TO<K07           MASK FOR LO BITS OF RESTORE
04F4  F0FD  FILE 068                 076     GETDS      BR IF LZ=0         BR IF RESTORE
04F6  0133  FILE 070  TSTIOA                            Z=U1*-K03          MASK FOR VALID DEVICE ADDR
04F8  F0FD  FILE 074                 076     GETDS      BR IF LZ=0         BR IF DA VALID
04FA  850E  FILE 075                 085     SETIVR     BR
04FC  5ECF  FILE 076  GETDS                             PO=DS              GET DISK STATUS
04FE  1CA3  FILE 077                                    PO=PO*-K0A         REMOVE TRAP GT, LO COMP
0500  C489  FILE 078                 082     BALMS      BR IF Z=0          BR IF NO UNSELECTED STATUS
0502  2C43  FILE 079                                    PO=0$K04           SET UNSELECTED STATUS BIT
0504  A5E6  FILE 080  STRSEN          273     SNSSTB     BAL                STORE SENSE
0506  855A  FILE 081                 123     STR04      BR
0508  8FA6  FILE 082  BALMS           316     MODSEL     BAL                BAL TO MOD SELECT
050A  EE95  FILE 083                 088     UNSAFE     BR IF DS2=1        BR IF UNSAFE
050C  DE99  FILE 084                 090     TSTDS7     BR IF DS1=1        BR IF ON LINE
050E  AEEC  FILE 085  SETIVR         362     ZROSEN     BAL                ZERO SENSE BYTES, INVALID 2311
0510  2A45  FILE 086                                    TO=0$K40           SET INT REQD
0512  8558  FILE 087                 122     CONCON     BR                 BR TO SET CONTINGENT CON
0514  2C85  FILE 088  UNSAFE                            PO=0$K80           SET UNSAFE FOF SENSE
0516  8504  FILE 089                 080     STRSEN     BR
0518  2E15  FILE 090  TSTDS7                            HO=0$K10           PRESET BUSY STATUS
051A  FAAB  FILE 091                 099     SKINC      BR IF DS7=1        BR IF SEEK INCOMPLETE
051C  CEBE  FILE 092                 109     BRANCH     BR IF DS0=0        BR IF NOT READY
051E  E5B7  FILE 093                 105     ATNRST     BR IF DASI2=1      BR IF SELECTED GATED ATTENTION
0520  F025  FILE 094                 096     TIO        BR IF GO BIT7=1    BR IF TIO
0522  9068  FILE 095                 149     CCTEST     BR                 BR TO CC TEST OF SIO ROUTINE
0524  49AF  FILE 096  TIO                                MS=TO              DESELECT FILE
0526  5CC2  FILE 097                                    RDH P DA,B8        RESTORE CPU P REG
0528  810C  FILE 098          CCOM 155     CCOB       BR                 SET CCO
052A  AEEC  FILE 099  SKINC           362     ZROSEN     BAL
052C  0A3B  FILE 100                                    Z=TO<K03           MASK FOR LO BITS OF RECAL
052E  F0C6  FILE 101                 113     RSTATN     BR IF LZNZ         BR IF NOT RECAL
0530  0A1D  FILE 102                                    Z=TO<K10           MASK FOR HI BITS RECAL
0532  F0C6  FILE 103                 113     RSTATN     BR IF HZNZ         BR IF NOT RECALIBRATE
0534  910C  FILE 104                 236     EXCUTE     BR
0536  A044  FILE 105  ATNRST          344     RSTATT     BAL                BAL TO RESET GATED ATTN
0538  2E43  FILE 106                                    HO=0$K04           SET DE BIT
053A  F03F  FILE 107                 109     BRANCH     BR IF GO BIT7=1    BR IF TEST I/O
053C  3E15  FILE 108                                    HO=HO$K10          OR IN BUSY STATUS FOR SIO
053E  A052  FILE 109  BRANCH          351     RSTMOD     BAL
0540  F061  FILE 110                 126     MODE       BR IF GO BIT7=1    BR IF TEST IO
0542  AEEC  FILE 111                 362     ZROSEN     BAL                BAL TO ZERO SENSE BYTES
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
0544  856A  FILE 112                 131    TSTPCI     BR
0546  2A13  FILE 113    RSTATN                          T0=0$K01           SET SEEK CHECK SENSE
0548  E5D6  FILE 114                 121    NOTDE      BR IF DAS I2=0     BR IF NO GATED ATTENTION
054A  A044  FILE 115                 344    RSTATT     BAL                BAL TO RESET GATED ATTENTION
054C  A5EC  FILE 116                 276    SNSSTC     BAL                BAL TO STORE SENSE
054E  3E43  FILE 117                                    H0=H0$K04          OR IN DEVICE END
0550  F05B  FILE 118                 123    STR04      BR IF GO BIT7=1    BR IF TEST I-O
0552  3E15  FILE 119                                    H0=H0$K10          OR IN BUSY BIT
0554  855A  FILE 120                 123    STR04      BR
0556  A044  FILE 121    NOTDE        344    RSTATT     BAL                BAL TO RESET GATED ATTENTION
0558  A5EC  FILE 122    CONCON       276    SNSSTC     BAL                STORE SENSE
055A  2B43  FILE 123    STR04                           T1=0$K04
055C  70AC  FILE 124                                    STH U AS,T+0
055E  F068  FILE 125    SIOTIO       130    TSTHIO     BR IF GO BIT7=0    BR IF NOT TIO
0560  2400  FILE 126    MODE                            SET MODE K=00      SET CPU MODE AND ZONE
0562  5CC2  FILE 127                                    RDH P DA,B8        RESTORE CPU P REG
0564  2F07  FILE 128                                    H1=0
0566  A01E  FILE 129                 CCOM 123   STATOS     BR
0568  F073  FILE 130    TSTHIO       135    ZERCHN     BR IF GO BIT6=1    BR IF HIO
056A  234B  FILE 131    TSTPCI                          V1=V1+K04          ADD 4 TO CCW ADDR
056C  5630  FILE 132                                    RDH D V+0
056E  2F85  FILE 133                                    H1=0$K80           SET PCI BIT FOR CHNL STATUS
0570  C275  FILE 134                 136    ZERT       BR IF D0 BIT4=1    BR IF PCI IN CURRENT CCW
0572  2F07  FILE 135    ZERCHN                          H1=0               ZERO CHANNEL STATUS
0574  2AC7  FILE 136    ZERT                            T0=0               ZERO HI ADDR BYTE
0576  C3FF  FILE 137                 141    BAON       BR IF BA4=1        BR IF IPL
0578  2400  FILE 138                                    SET MODE K=00      SET CPU MODE AND ZONE
057A  5CC2  FILE 139                                    RDH P DA,B8        RESTORE CPU P REG
057C  A030  FILE 140                 CCOM 135   STATUS     BR              BR TO STORE SRATUS AND SET CC1
057E  A7D8  FILE 141     BAON        DCLL 026   STOP       BR              BR TO IPL STOP
            FILE 142    AEND
            FILE 143    *          ************************************************************************
            FILE 144    *          *                                                                    **
            FILE 145    *          *                  CYCLIC CODE HARDWARE TEST                         **
            FILE 146    *          *                                                                    **
            FILE 147    *          ************************************************************************
            FILE 148    ASEQ    AL07=68
1068  AEEC  FILE 149     CCTEST              362    ZROSEN     BAL                CLEAR THE SENSE BUFFER
106A  3E00  FILE 150                                    SET FIB K=80       SET NTO LATCH
            FILE 151    *
            FILE 152    *                    NOTE.
            FILE 153    *                    CHANGE THE FOLLOWING INSTR TO AN UNCOND BR TO -EXCUTE-
            FILE 154    *                    TO BYPASS THE CC TEST WHEN SCOPING.
            FILE 155    *
106C  2490  FILE 156                                    SET MODE K=09      SET FILE MODE AND ZONE
106E  0E08  FILE 157                                    RST FIB K=40       INITIAL RESET
1070  2302  FILE 158                                    SET DIAC K=10      SET DIAGNOSTIC MODE
1072  2304  FILE 159                                    SET DIAC K=20      SET DIAG INDEX
1074  3110  FILE 160                                    SET DIAB K=81      SET DIAG ADDR 0
1076  3D29  FILE 161                                    P1=0-K20           SET CONSTANT DF
1078  4FDF  FILE 162                                    FOP=P1             LOAD OP REG
```

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
107A	43DF	FILE 163				FEBO=P1	LOAD WRITE BUFFER
107C	8678	FILE 164		252	PEDS	BAL	TEST FOR CORRECT BITS IN WR BUFF
107E	5FCF	FILE 165				PO=FOP	GET OP REG BITS
1080	1CFF	FILE 166				PO=PO<KFF	MASK FOR CORRECT OP REG BITS
1082	C4FC	FILE 167		228	ALUCHK	BR IF ZNZ	BR IF OP REG ERROR
1084	4FCF	FILE 168				FOP=PO	RESET OP REG
1086	5FCF	FILE 169				PO=FOP	GET OP REG BITS
1088	C4FC	FILE 170		228	ALUCHK	BR IF ZNZ	BR IF OP REG NOT RESET
108A	866C	FILE 171		246	SHIFT4	BAL	ADVANCE
108C	866C	FILE 172		246	SHIFT4	BAL	BIT RING
108E	5BCF	FILE 173				PO=SDI	GET READ BUFF BITS
1090	867A	FILE 174		253	XOR	BAL	TST XFER OF WR TO RD BUFF
1092	5ECF	FILE 175				PO=DS	
1094	C4FC	FILE 176		228	ALUCHK	BR IF ZNZ	BR IF WR BUFF NOT RESET
1096	2310	FILE 177				SET DIAC K=01	ADV TO ZONE 1
1098	2390	FILE 178				SET DIAC K=09	ADV TO ZONE 2&SET SEP DATA
109A	2D13	FILE 179				P1=0$K01	SET WRITE BIT
109C	4FDF	FILE 180				FOP=P1	LOAD OP REG
109E	2D25	FILE 181				P1=0$K20	
10A0	43DF	FILE 182				FEBO=P1	LOAD WR BUFF
10A2	3110	FILE 183				SET DIAB K=81	SET READ & COMPARE GATES
10A4	8670	FILE 184		248	SHIFT2	BAL	ADV BIT RING
10A6	2D1F	FILE 185				P1=P1+K11	P1=31
10A8	4FDF	FILE 186				FOP=P1	LOAD OP REG
10AA	866C	FILE 187		246	SHIFT4	BAL	ADVANCE
10AC	8670	FILE 188		248	SHIFT2	BAL	BIT RING
10AE	5BCF	FILE 189				PO=SDI	GET READ BUFF BITD
10B0	2D85	FILE 190				P1=0$K80	
10B2	3D23	FILE 191				P1=P1$K02	
10B4	867A	FILE 192		253	XOR	BAL	TST FOR CORRECT WR BUFF SHUFT
10B6	3D73	FILE 193				P1=P1$K07	
10B8	8676	FILE 194		251	SET3	BAL	TST FOR CORRECT BITS IN CC REG
10BA	3114	FILE 195				SET DIAB K=A1	SET READ & COMPARE GATES
10BC	866C	FILE 196		246	SHIFT4	BAL	ADV BIT RING
10BE	2DA3	FILE 197				P1=0$K0A	
10C0	8676	FILE 198		251	SET3	BAL	TST FOR CORRECT CC DATA
10C2	1D00	FILE 199				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
10C4	CAFD	FILE 200		228	ALUCHK	BR IF DS4=1	BR IF UNEQUAL NOT RESET
10C6	2380	FILE 201				SET DIAC K=08	SET SEPERATED DATA
10C8	3114	FILE 202				SET DIAB K=A1	SET READ & COMPARE GATES
10CA	866C	FILE 203		246	SHIFT4	BAL	ADV BIT RING
10CC	3D47	FILE 204				P1=P1$K44	P1=4E
10CE	8676	FILE 205		251	SET3	BAL	TEST FOR CORRECT CC DATA
10D0	1D00	FILE 206				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES
10D2	2390	FILE 207				SET DIAC K=09	ADV TO ZONE 3&SET SEP DATA BIT
10D4	3114	FILE 208				SET DIAB K=A1	SET READ & COMPARE GATES
10D6	2308	FILE 209				SET DIAC K=40	ADV BIT RING
10D8	2D75	FILE 210				P1=0$K70	
10DA	3DA3	FILE 211				P1=P1$K0A	
10DC	8676	FILE 212		251	SET3	BAL	TEST FOR CORRECT CC DATA
10DE	1D00	FILE 213				RST FIA K=80	RST HI LO CC ERR&TRAP LATCHES

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
10E0    3114    FILE 214                                    SET DIAB K=A1      SET READ & COMPARE GATES
10E2    2380    FILE 215                                    SET DIAC K=08      SET SEP DATA BIT
10E4    2308    FILE 216                                    SET DIAC K=40      ADV BIT RING
10E6    1D45    FILE 217                                    P1=P1*-K40         P1=3A
10E8    8676    FILE 218                  251   SET3        BAL                TEST FOR CORRECT CC DATA
10EA    1D00    FILE 219                                    RST FIA K=80       RST HI LO CC ERR&TRAP LATCHES
10EC    3114    FILE 220                                    SET DIAB K=A1      SET READ & COMPARE GATES
10EE    2308    FILE 221                                    SET DIAC K=40      ADV BIT RING
10F0    2D55    FILE 222                                    P1=0$K50
10F2    3DB3    FILE 223                                    P1=P1$K0B
10F4    8676    FILE 224                  251   SET3        BAL                TST FOR CORRECT CC DATA
10F6    1D00    FILE 225                                    RST FIA K=80       RST HI LO CC ERR&TRAP LATCHES
10F8    2480    FILE 226                                    SET MODE K=08      SET FILE MODE CPU ZONE
10FA    9108    FILE 227                  234   CCNOER      BR                 END OF TEST - RESULTS OK
10FC    2100    FILE 228    ALUCHK                          SET DIAB K=00      RST DIAG ADDR LATCHES
10FE    1E00    FILE 229                                    RST FIB K=80       CHAIN END RESET
1100    0E08    FILE 230                                    RST FIB K=40       INITIAL RST
1102    2480    FILE 231                                    SET MODE K=08      SET FILE MODE-CPU ZONE
1104    3C83    FILE 232                                    P0=P0$K08          SET ALU CHECK SENSE BIT
1106    8504    FILE 233                  080   STRSEN      BR
1108    2100    FILE 234    CCNOER                          SET DIAB K=00      RESET DAAG ADDR LATCHES
110A    3E00    FILE 235                                    SET FIB K=80       SET NTO LATCH
110C    0E08    FILE 236    EXCUTE                          RST FIB K=40       INITIAL RST
110E    0E04    FILE 237                                    RST FIB K=20       CLOLD START
1110    2D43    FILE 238                                    P1=0$K04           SET HD SELECT BIT
1112    4EDF    FILE 239                                    FBO=P1             HD SELECT TO BUSS
1114    5CC2    FILE 240                                    RDH P DA,B8        RESTORE CPU P REG
1116    2A07    FILE 241                                    TO=0
1118    2490    FILE 242                                    SET MODE K=09      SET FILE MODE AND ZONE
111A    5F42    FILE 243                                    RDH H DA,98        RD OUT CCW NA
111C    831A    FILE 244                FILT 038  GETOP     BR                 BR TO SIO,TRAP COMMON WORD
                FILE 245    AEND
066C    2308    FILE 246     SHIFT4                         SET DIAC K=40
066E    2308    FILE 247     SHIFT3                         SET DIAC K=40
0670    2308    FILE 248     SHIFT2                         SET DIAC K=40
0672    2308    FILE 249     SHIFT1                         SET DIAC K=40
0674    128E    FILE 250     RETURN                         RTN
0676    2112    FILE 251     SET3                           SET DIAB K=11      SET DIAG ADDR 3
0678    5ECF    FILE 252     PEDS                           P0=DS
067A    6CD1    FILE 253     XOR                            P0=P0<P1           MASK FOR CORRECT BITS
067C    C4F5    FILE 254                  250   RETURN      BR IF Z=0          BR IF NO ERROR
067E    90FC    FILE 255                  228   ALUCHK      BR
                FILE 256    *             ********************************************************************
                FILE 257    *             *                 CONTINGENT CONNECTION TEST - SUBROUTINE          **
0224    5C52    FILE 258     TSTADR                         RDH P DA,9A        GET SNS BYTES 0&1
0226    F939    FILE 259                  268   PREST       BR IF P1 BIT7=1    BR IF NONSENSE STORED
0228    EC38    FILE 260                  268   PREST       BR IF P0 BIT2=0    BR IF NO UNSENSED SENSE
022A    2C07    FILE 261                                    P0=0                     SET AUX
022C    2D43    FILE 262                                    P1=0$K04               ADDR 0004
022E    5EC0    FILE 263                                    RDH H AS,P+0       GET I/O ADDR
0230    4FCF    FILE 264                                    FOP=P0             ZERO OP REG
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
0232    6F11    FILE 265                                   H1=H1<U1           COMPARE I/O ADDRESSES
0234    C4B9    FILE 266                 268     PREST     BR IF Z=0          BR IF COMPARE EQUAL
0236    4FDF    FILE 267                                   FOP=P1             SET OP REG INDICATOR
0238    5CC2    FILE 268    PREST                          RDH P DA,B8        RESTORE CPU P REG
023A    128E    FILE 269                                   RTN
                FILE 270    *              *                SET SENSE BYTES          - SUBROUTINE              **
                FILE 271    *              ***************************************************************************
25E4    2C15    FILE 272    SNSSTA                         P0=0$K10
25E6    2D07    FILE 273    SNSSTB                         P1=0
25E8    7C62    FILE 274                                   STH P DA,9C        STORE SENSE BYTES 2,3
25EA    2A15    FILE 275                                   T0=0$K10
25EC    3A25    FILE 276    SNSSTC                         T0=T0$K20          SET CONTINGENT CONN IND.SENSE0-2
25EE    2B07    FILE 277                                   T1=0
25F0    7A52    FILE 278                                   STH T DA,9A        STORE SENSE BYTES 0,1
25F2    2A07    FILE 279                                   T0=0
25F4    2E23    FILE 280                                   H0=0$K02           SET UNIT CHK DEVICE STATUS
25F6    49BF    FILE 281                                   MS=T1              DESELECT MOD
25F8    128E    FILE 282                                   RTN
                FILE 283    *              ***************************************************************************
                FILE 284    *              *                                                                         **
                FILE 285    *              *                SET DEVICE ADDRESS       - SUBROUTINE              **
                FILE 286    *              *                FOR GATED ATTENTION                                 **
                FILE 287    *              ***************************************************************************
0F90    2D95    FILE 288    SETDA                          P1=0$K90           SET DEVICE ADDR 0
0F92    0E08    FILE 289                                   RST FIB K=40       GIVE INITL RESET TO RST CHN END
0F94    2100    FILE 290                                   SET DIAB K=00
0F96    CCAB    FILE 295                 318     MODBR     BR IF FGA0=1       BR IF MOD 0 GATED ATTENTION
0F98    2D1B    FILE 296                                   P1=P1+K01
0F9A    DCAB    FILE 297                 318     MODBR     BR IF FGA1=1       BR IF MOD 1 GATED ATTENTION
0F9C    2D1B    FILE 298                                   P1=P1+K01
0F9E    ECAB    FILE 299                 318     MODBR     BR IF FGA2=1       BR IF MOD 2 GATED ATTENTION
0FA0    2D1B    FILE 300                                   P1=P1+K01
0FA2    FCAB    FILE 301                 318     MODBR     BR IF FGA3=1       BR IF MOD 3 GATED ATTENTION
0FA4    128F    FILE 305                                   RTN
                FILE 311    *              ***************************************************************************
                FILE 312    *              *                                                                         **
                FILE 313    *              *                  MODULE SELECT          - SUBROUTINE              **
                FILE 314    *              *                                                                         **
                FILE 315    *              ***************************************************************************
0FA6    5C32    FILE 316    MODSEL                         RDH P DA,8E        GET I/O ADDR
0FA8    0E08    FILE 317                                   RST FIB K=40       GIVE INITL RESET TO RST CHN END
0FAA    2E07    FILE 318    MODBR                          H0=0
0FAC    C90B    FILE 319                 336     ENDSEL    BR IF P1 BIT4=1    BR IF DRIVE IS 8 OR HIGHER
0FAE    D90B    FILE 321                 336     ENDSEL    BR IF P1 BIT5=1    BR IF DRIVE IS 4 OR HIGHER
0FB0    2C15    FILE 323                                   P0=0$K10           P0=10
0FB2    891F    FILE 324                 325     MOD    N  N=P1 BITS67        BR ON DRIVE ADDRESS
0F80    6CC3    FILE 325    MOD    0                       P0=P0+P0           FORM
0F82    6CC3    FILE 326    MOD    1                       P0=P0+P0              DESIRED
0F84    6CC3    FILE 327    MOD    2                       P0=P0+P0                 DRIVE MASK
0F86    4BEF    FILE 328    MOD    3                       TGR0=H0            RESET TAG REG
0F88    49CF    FILE 334                                   MS=P0              MODULE SELECT
```

```
ADDR    WORD  SEQUENCE NO.  LABEL    NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
0F8A    4EEF    FILE 336    ENDSEL                          FBO=H0             RESET FILE BUS OUT
0F8C    4DEF    FILE 337                                    FFO=H0             RESET CCW FILE FLAGS
0F8E    128E    FILE 338                                    RTN
                FILE 339  *          ************************************************************
                FILE 340  *          *                                                          **
                FILE 341  *          *              RESET GATED ATTENTION     - SUBROUTINE      **
                FILE 342  *          *                                                          **
                FILE 343  *          ************************************************************
2044    2C07    FILE 344    RSTATT                          P0=0               RESET GATED ATTENTION SUBR
2046    4BCF    FILE 345                                    TGRO=P0            RESET CONTROL
2048    4ECF    FILE 346                                    FBO=P0             RESET BUSS
204A    2D15    FILE 347                                    P1=0$K10
204C    4BDF    FILE 348                                    TGRO=P1            RAISE CONTROL
204E    2D45    FILE 349                                    P1=0$K40           RD BIT 5
2050    4EDF    FILE 350                                    FBO=P1             RAISE RD BIT,RST GATED ATTN
2052    2D07    FILE 351    RSTMOD                          P1=0
2054    4BDF    FILE 352                                    TGRO=P1            RESET CONTROL
2056    4EDF    FILE 353                                    FBO=P1             RESET FILE BUS
2058    4FDF    FILE 354                                    FOP=P1             RESET FILE OP REG
205A    49DF    FILE 355                                    MS=P1              DESELECT MOD
205C    128E    FILE 356                                    RTN
                FILE 357  *          ************************************************************
                FILE 358  *          *                                                          **
                FILE 359  *          *              ZERO SENSE BYTES          - SUBROUTINE      **
                FILE 360  *          *                                                          **
                FILE 361  *          ************************************************************
2EEC    4C52    FILE 362    ZROSEN                          RDH P DC,9A        P=0000
2EEE    7C62    FILE 363                                    STH P DA,9C        ZERO SENSE BYTES 2,3
2EF0    5CC4    FILE 364                                    P=P+1              P=0001
2EF2    7C52    FILE 365                                    STH P DA,9A        ZERO SENSE BYTES 0,1
2EF4    128E    FILE 366                                    RTN
                                     ************************************
                                     * CROSS REFERENCE FOR CSECT FILE   *
                                     ************************************

FILE 008    DCLA 125
FILE 019    FILE 011
FILE 026    FILE 012  FILE 013  FILE 015
FILE 027    FILE 021  FILE 033
FILE 031    FILE 010
FILE 040    FILE 027  FILE 034  FILE 045  FILE 062
FILE 042    FILE 029
FILE 043    FILE 035
FILE 048    FILE 043
FILE 051    FILE 049
FILE 053    FILE 036
FILE 060    FILE 057
FILE 063    FILE 060
FILE 070    FILE 066
FILE 076    FILE 037  FILE 039  FILE 064  FILE 068  FILE 074
FILE 080    FILE 089  FILE 233
FILE 082    FILE 078
```

```
                                    ************************************
                                    * CROSS REFERENCE FOR CSECT FILE *
                                    ************************************

FILE 085    FILE 075
FILE 088    FILE 083
FILE 090    FILE 084
FILE 096    FILE 094
FILE 099    FILE 091
FILE 105    FILE 093
FILE 109    FILE 092  FILE 107
FILE 113    FILE 101  FILE 103
FILE 121    FILE 114
FILE 122    FILE 087
FILE 123    FILE 081  FILE 118  FILE 120
FILE 125    FILE 042  FILE 052
FILE 126    FILE 110
FILE 130    FILE 125
FILE 131    FILE 112
FILE 135    FILE 130
FILE 136    FILE 134
FILE 141    FILE 137
FILE 149    FILE 095
FILE 228    FILE 167  FILE 170  FILE 176  FILE 200  FILE 255
FILE 234    FILE 227
FILE 236    FILE 059  FILE 104
FILE 246    FILE 171  FILE 172  FILE 187  FILE 196  FILE 203
FILE 248    FILE 184  FILE 188
FILE 250    FILE 254
FILE 251    FILE 194  FILE 198  FILE 205  FILE 212  FILE 218  FILE 224
FILE 252    FILE 164
FILE 253    FILE 174  FILE 192
FILE 258    FILE 038  FILE 044  FILE 061
FILE 268    FILE 259  FILE 260  FILE 266
FILE 272    FINT 025
FILE 273    FILE 080
FILE 276    FILE 116  FILE 122  FINT 034
FILE 288    BSYS 043  FINT 027
FILE 316    FILE 058  FILE 082
FILE 318    FILE 295  FILE 297  FILE 299  FILE 301
FILE 325    FILE 324
FILE 336    FILE 319  FILE 321
FILE 344    BSYS 044  FILE 105  FILE 115  FILE 121  FINT 031  FINT 033
FILE 351    FILE 109
FILE 362    FILE 085  FILE 099  FILE 111  FILE 149
```

ENTRY POINTS

(0140) THIS IS THE ASSIGNED ADDRESS FOR TRAP ENTRY THE OPERATION BRANCHES ON THE CHAINING, STATUS MODIFIER, AND UNUSUAL CONDITION INDICATORS. THE ROUTINE EITHER READS IN THE NEXT CCW OR BRANCHES TO THE ENDING SEQUENCE.

GETOP THIS IS THE NORMAL ENTRY FOR THE INITIAL SEQUENCE TO OBTAIN THE FIRST CCW. THE CCW IS TESTED FOR VALID FLAGS AND COUNT BEFORE BRANCHING FOR A COMMAND DECODE.

MCHCHK THIS IS A RETURN ENTRY TO THE ROUTINE USED AFTER BRANCHING TO THE MACHINE CHECK TRAP ROUTINE.

PTRCHK THIS ENTRY IS FROM THE BWRP ROUTINE TO SET THE PROTECTION CHECK STATUS BIT.

WRAP THIS ENTRY IS FROM THE BWRP ROUTINE TO SET THE PROGRAM CHECK STATUS BIT.

IPL THIS ENTRY IS MADE DURING THE IPL OPERATION. THE OPERATION LOOPS ON THE FIRST COMMAND UNTIL THE DE TRAP FOR THE RECALIBRATE CLEARS THE LOOP CONDITION. THE OPERATION CONTINUES WITH THE READ DATA SEQUENCE FOR THE INITIAL 24 BYTES.

SELSTA THIS ENTRY IS USED TO SET THE EQUIPMENT SENSE BIT WHEN AN OP REGISTER FAILURE IS DETECTED.

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS
                FILT 001    T                                2311 FEATURE    TRAP ROUTINE FOR FILES
                FILT 002    ATABLE    ADDR=0140
0140    2222    FILT 004                                     LINK V MMSK1=1         SET FILE PRIORITY-STORE BACK UP
0142    8D87    FILT 018     FILE                 025  TRAP   N  N=FFI BITS23
                FILT 024    AEND
0300    96A2    FILT 025     TRAP     0          086  NOTCHN     BR                  UNUSUAL CONDITION OR NO CHAINING
0302    96A2    FILT 026     TRAP     1          086  NOTCHN     BR                  STAT MOD AND UNUS COND OR NO CHN
0304    830A    FILT 027     TRAP     2          030  FRSTBT     BR                  CMND OR DATA CHAIN-NO UNUS COND
0306    2F7B    FILT 028     TRAP     3                          H1=H1+K07           STATUS MOD AND CMND OR DATA CHN
0308    5EE4    FILT 029                                        H=H+1               ADD 8 TO CCW ADDRESS
030A    1D0C    FILT 030     FRSTBT                             RST FIA K=80        RESET THE TRAP LATCH
030C    5CF8    FILT 031                                        RDH P H+2           GET CCW BYTES 0&1
030E    A967    FILT 032                          079  PROCHK N   BR IF P1=NZ         BR IF 2ND BYTE INVALID
0310    0C8B    FILT 033                                        Z=P0<K08            MASK FOR TIC
0312    F0A2    FILT 034                          042  NOTTIC     BR IF LZNZ          BR IF NOT TIC
0314    5EF0    FILT 035                                        RDH H H+0           GET NEW CCW ADDRESS
0316    0F83    FILT 036                                        Z=H1*-K08           MASK FOR DOUBLE WORD BOUNDRY
0318    F0DE    FILT 037                          078  TICCHK     BR IF LZNZ          BR IF INVALID ADDRESS
031A    5CF8    FILT 038     GETOP                              RDH P H+2           GET OP
031C    A967    FILT 039                          079  PROCHK N   BR IF P1=NZ
031E    0C8B    FILT 040                                        Z=P0<K08
0320    C4E1    FILT 041                          079  PROCHK O   BR IF Z=0
0322    CDA9    FILT 042     NOTTIC               045  CDBR       BR IF FFIO=1        BR IF DATA CHAIN
0324    4FCF    FILT 043                                        FOP=P0              LOAD THE OP REG
0326    F0E1    FILT 044                          079  PROCHK O   BR IF LZ=0          BR IF INVALID OP
0328    5AF8    FILT 045     CDBR                               RDH T H+2           GET DATA ADDR FROM CCW
032A    5CF8    FILT 046                                        RDH P H+2
```

```
ADDR   WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
032C   58F8    FILT 047                                    RDH I H+2          GET CCW BYTES 6&7
032E   0C83    FILT 048                                    Z=P0*-K08          MASK FOR VALID FLAG
0330   F0E4    FILT 049                 081   BUGBUG       BR IF LZNZ         BR ON INVALID FLAG BITS 567 NZ
0332   4886    FILT 050                                    I=I                TEST FOR VALID CCW COUNT
0334   C4E5    FILT 051                 081   BUGBUG       BR IF Z=0          BR IF INVALID COUNT OF 0000
0336   7E42    FILT 052                                    STH H DA,98        SAVE CCW ADDR
0338   FC3E    FILT 053                 057   SKOFF        BR IF P0 BIT3=0    TEST FOR SKIP BIT OFF
033A   FBBF    FILT 054                 057   SKOFF        BR IF FOP7=1       SKIP THE SKIP BIT ON WRITE OP
033C   2A07    FILT 055                                    T0=0               ZERO UPPER DATA ADDR TO PREVENT
               FILT 056  *                                                     MEMORY WRAP IF SKIP BIT IS ON
033E   CDED    FILT 057   SKOFF         061   DATCHN       BR IF FFIO=1       BR IF DATA CHAIN
0340   4DCF    FILT 058                                    FF0=P0             SET CCW FLAGS IN HARDWARE
0342   5C52    FILT 059                                    RDH P DA,9A        GET PREVIOUS OP AND FILE MASK
0344   BB9D    FILT 060           FILX 007   DECODE N     N=FOPL             BR TO DECODE THE OP
036C   4DCF    FILT 061   DATCHN                           FF0=P0             SET FLAGS IN HDWR
036E   2C07    FILT 062                                    P0=0                     SET
0370   2D35    FILT 063                                    P1=0$K30               AUX ADDRESS
0372   3D43    FILT 064                                    P1=P1$K04                0034
0374   5ECC    FILT 065                                    RDH H AS,P+0       GET ADDR OF NEXT BYTE-SK SNS DC
0376   8BA3    FILT 066                 067   CDRTN  N     N=FOP BITS67
0120   9852    FILT 067   CDRTN  0  FILX 261   LASTST       BR                 SENSE OP  TEST FOR LAST BYTE
0122   8132    FILT 068   CDRTN  1        076   RDWR         BR
0124   8132    FILT 069   CDRTN  2        076   RDWR         BR
0126   FFAE    FILT 070   CDRTN  3        074   TST5         BR IF FOP3=0       CTL OP-BR IF SEEK
0128   DBA0    FILT 071                 067   CDRTN  0     BR IF FOP5=0       BR IF SEEK
012A   5C52    FILT 072                                    RDH P DA,9A        GET FILE MASK BYTE
012C   860E    FILT 073           FILX 097   FMRETN       BR                 RETURN TO SET FILE MASK
012E   DBA0    FILT 074   TST5          067   CDRTN  0     BR IF FOP5=0
0130   CBA0    FILT 075                 067   CDRTN  0     BR IF FOP4=0
0132   5E42    FILT 076   RDWR                             RDH H DA,98        GET CCW ADDR
0134   8600    FILT 077           FILX 213   CDEND        BR
035E   5E42    FILT 078   TICCHK                           RDH H DA,98        GET CCW ADR
0360   3F73    FILT 079   PROCHK 0                         H1=H1$K07          OR IN CCW BITS TO UPDATE ADDR
0362   5EE4    FILT 080                                    H=H+1              CCW ADDR UPDATE COMPLETE
0364   7E42    FILT 081   BUGBUG                           STH H DA,98        SAVE THE CCW ADDRESS
0366   2F25    FILT 082                                    H1=0$K20           SET PROGRAM CHECK FOR STATUS
0368   2E07    FILT 083                                    H0=0               ZERO DEVICE STATUS
036A   92BA    FILT 084           FILX 439   ZROSN2       BR
               FILT 085  ASEQ    AL07=22
16A2   50FF    FILT 086   NOTCHN                           H1=TGRI            GET CPU DETECTED ERRORS
16A4   0F83    FILT 087                                    Z=H1*-K08          CHK FOR CPU DETECTED FILE ERROR
16A6   F0BD    FILT 088                 099   NOCPUE       BR IF LZ=0         BR IF NO CPU DETECTED FILE ERROR
16A8   DB2E    FILT 089                 092   TSTPRT       BR IF H1 BIT5=0    BR IF NOT MACHINE CHECK
16AA   8220    FILT 090           BMCK 003   TRAPWD       BR                    STORE LOG OUT AND RETURN
16AC   2F43    FILT 091   MCHCHK                           H1=0$K04           SET CHAN CTL CHK FOR STATUS
16AE   EB32    FILT 092   TSTPRT        094   TSTWRP       BR IF H1 BIT6=0    BR IF NOT STORAGE PROTECT
16B0   2F15    FILT 093   PRTCHK                           H1=0$K10           SET PROTECTION CHK FOR STATUS
16B2   FB36    FILT 094   TSTWRP        096   RSTBC        BR IF H1 BIT7=0    BR IF NOT STORAGE WRAP
16B4   2F25    FILT 095   WRAP                             H1=0$K20           SET PROGRAM CHK FOR STATUS
16B6   0680    FILT 096   RSTBC                           RST BC K=08        RST CPU ERROR LATCHES FOR FILE
16B8   2EC3    FILT 097   DEVSTA                          H0=0$K0C           SET CE+DE FOR STATUS
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
16BA    977E    FILT 098                  196     TSTSM      BR
16BC    2F07    FILT 099    NOCPUE                           H1=0
16BE    C1B8    FILT 100                  097     DEVSTA     BR IF DASI4=0      BR IF NO UNUSUAL CONDITIONS
16C0    E5FE    FILT 101                  132     NOTGAT     BR IF DASI2=0      BR IF NOT SEL GATED ATTENTION
16C2    2D45    FILT 102                                     P1=0$K40           SET READ GATE BIT
16C4    4EDF    FILT 103                                     FBO=P1             PUT RD GATE ON BUSS TO FILE
16C6    2E15    FILT 104                                     H0=0$K10           SET CTL TAG BIT
16C8    4BEF    FILT 105                                     TGRO=H0            ISSUE CTL PULSE
16CA    2E43    FILT 106                                     H0=0$K04           SET SELECT HEAD BIT
16CC    4BFF    FILT 107                                     TGRO=H1            RESET TAGS
16CE    1D00    FILT 108                                     RST FIA K=80       RESET TRAP LATCH
16D0    4EEF    FILT 109                                     FBO=H0             PUT HD SEL ON BUSS
16D2    FBD6    FILT 110                  112     TSTBA      BR IF FOP7=0       BR IF FROM IPL
16D4    973C    FILT 111                  163     INSERT     BR                 GO TEST FOR SEEK INCOMPLETE
16D6    C3FD    FILT 112    TSTBA         131     RTNWD      BR IF BA4=1        BR IF IPL LATCH IS ON
16D8    FD59    FILT 113    IPL           113     IPL        BR IF P1 BIT3=1    WAIT FOR DEVICE END FROM RECAL
16DA    92F6    FILT 114                FILX 471  EXAMIN     BR                 GO TEST FOR WHOLESOME RECAL
16DC    6EE6    FILT 115    GOCAT                            H=H-2              DECREMENT CCW ADDR
16DE    6EE6    FILT 116                                     H=H-2                    TO FLAG BYTE
16E0    5CF8    FILT 117                                     RDH P H+2          GET FLAGS
16E2    4DCF    FILT 118                                     FFO=P0             SET FLAGS IN FLAG REG
16E4    2C63    FILT 119                                     P0=0$KC6           SET READ DATA BITS
16E6    4FCF    FILT 120                                     FOP=P0             SET READ DATA IN OP REG
16E8    5C52    FILT 121                                     RDH P DA,9A        GET PREVIOUS OP IND
16EA    2C07    FILT 122                                     P0=0               SET PREV OP
16EC    F8EC    FILT 123    WTEIND        123     WTEIND     BR IF FGA7=0       WAIT FOR INDEX
16EE    7C52    FILT 124                                     STH P DA,9A        STORE PREVIOUS OP
16F0    0E04    FILT 125                                     RST FIB K=20       COLD START RESET
16F2    3D00    FILT 126                                     SET FIA K=80       ISSUE GO PULSE
16F4    2C15    FILT 127                                     P0=0$K10           SET CONTROL BIT
16F6    4BCF    FILT 128                                     TGRO=P0            SET CTL TAG
16F8    6EE4    FILT 129                                     H=H+2              RESTORE CCW ADDR
16FA    C3FB    FILT 130    WTEBA         130     WTEBA      BR IF BA4=1        IF IPL LATCH ON WAIT HERE
16FC    0222    FILT 131    RTNWD                            RTN V MMSK1=0      RETURN FROM TRAP
16FE    CA86    FILT 132    NOTGAT        136     TSTTC      BR IF DS4=0        BR IF NOT TRAP GATE
1700    DF87    FILT 133                  136     TSTTC      BR IF FOP1=1       BR IF MULTI-TRACK BIT ON
1702    4BFF    FILT 134                                     TGRO=H1            RESET TAG REG
1704    870E    FILT 135                FILX 321  TRPRTN     BR
1706    5ADF    FILT 136    TSTTC                            P1=TC              GET TERMINATING CONDITIONS
1708    C4C0    FILT 137                  165     ERRTST     BR IF ZNZ          BR IF ANY ERRORS
170A    5C62    FILT 138                                     RDH P DA,9C        GET BUFFERED SEEK DATA
170C    CA9F    FILT 139                  148     SWITHD     BR IF DS4=1        BR IF HEAD SWITCH TIME
170E    C5C0    FILT 140                  165     ERRTST     BR IF DASI0=0      BR IF NOT COMPARE HA
1710    5BCF    FILT 141                                     P0=SDI             GET HA FROM READ BUFFER
1712    6CD1    FILT 142                                     P0=P0<P1           COMPARE HEAD NUMBERS
1714    F098    FILT 143                  145     NOCOMP     BR IF LZNZ         BR IF AT WRONG HEAD
1716    928C    FILT 144                FILX 416  SETCTL     BR
1718    2D07    FILT 145    NOCOMP                           P1=0               ZERO SENSE BYTE 1
171A    2C13    FILT 146    SKCHK                            P0=0$K01           SET SEEK CHK FOR SENSE
171C    976E    FILT 147                  188     CHANZ      BR
171E    5E52    FILT 148    SWITHD                           RDH H DA,9A        GET FILE MASK ALG
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS
1720    EB31    FILT 149                   157  FIPRO     BR IF H1 BIT6=1    BR IF FILE PROTECTED
1722    CD32    FILT 150                   158  NOSEEK    BR IF P1 BIT0=0    BR IF NO PREV SEEK
1724    2D1B    FILT 151                                  P1=P1+K01          ADD 1 TO HEAD NUMBER
1726    7C62    FILT 152                                  STH P DA,9C        STORE NEW HD
1728    5DF9    FILT 153                                  H1=P1
172A    0DAB    FILT 154                                  Z=P1<K0A           MASK FOR HD 10
172C    F0B5    FILT 155                   159  HDSEOC    BR IF LZ=0         BR IF END OF CYL-HD 10
172E    9254    FILT 156              FILX 388  GETCYL    BR
1730    876A    FILT 157  FIPRO       FILX 361  PROTEC    BR
1732    859A    FILT 158  NOSEEK      FILX 162  INVSEQ    BR
1734    2D25    FILT 159  HDSEOC                          P1=0$K20           SET END OF CYL FOR SENSE
1736    2C07    FILT 160                                  P0=0               ZERO SENSE BYTE 0
1738    2F07    FILT 161                                  H1=0               ZERO CHAN STATUS
173A    976E    FILT 162                   188  CHANZ     BR
173C    FAC1    FILT 163  INSERT           165  ERRTST    BR IF DS BIT7=1    TEST FOR SEEK INCOMPLETE
173E    8640    FILT 164              FILX 122  NTORST    BR                   OBTAIN NEXT CCW
1740    2D07    FILT 165  ERRTST                          P1=0
1742    2C45    FILT 166                                  P0=0$K40           PRESET INTERVENTION REQ SNS
1744    DEEE    FILT 167                   188  CHANZ     BR IF DS1=0        BR IF NOT READY
1746    2E85    FILT 168                                  H0=0$K80           PRESET UNSAFE SENSE
1748    EEDD    FILT 169                   179  EQUCHK    BR IF DS2=1        BR IF UNSAFE
174A    FA9B    FILT 170                   146  SKCHK     BR IF DS7=1        BR IF SEEK CHECK
174C    5ACF    FILT 171                                  P0=TC              GET TERMINATING CONDITIONS
174E    C4E0    FILT 172                   181  SETSNS    BR IF ZNZ          BR IF ANY ERRORS
1750    7C62    FILT 173                                  STH P DA,9C        STORE SNS BYTE 2
1752    2EC3    FILT 174                                  H0=0$K0C           SET CE+DE FOR STATUS
1754    D8F9    FILT 175                   193  SETUEX    BR IF FGA5=1       BR IF UNIT EXCEPTION
1756    C8FB    FILT 176                   194  STR9A     BR IF FGA4=1       BR IF ICL
1758    2D07    FILT 177  SELSTA                          P1=0
175A    2E15    FILT 178                                  H0=0$K10           SET U SELECTED STATUA FOR SNS
175C    2C15    FILT 179  EQUCHK                          P0=0$K10           SET EQUIPMENT CHF FOR SNS
175E    9770    FILT 180                   189  STR9C     BR
1760    5CDB    FILT 181  SETSNS                          P1=P0H             SET SENSE BYTE 1
1762    1CF5    FILT 182                                  P0=P0*-KF0         STRIP EXTRA BITS FROM SNS BYTE 0
1764    FD68    FILT 183                   185  TSTMAM    BR IF P1 BIT2=0    BR IF NOT NO REC FOUND
1766    3D83    FILT 184                                  P1=P1$K08          SET NO REC FOUND SENSE
1768    FD6C    FILT 185  TSTMAM           187  STRIP     BR IF P1 BIT3=0    BR IF NOT MISSING ADDR MARK
176A    3D23    FILT 186                                  P1=P1$K02          SET MISS AM FOR SENSE
176C    1D35    FILT 187  STRIP                           P1=P1*-K30         STRIP EXTRA BITS FROM SNS BYTE 1
176E    2E07    FILT 188  CHANZ                           H0=0               ZERO SNS BTTE 2&CHAN STATUS
1770    7E62    FILT 189  STR9C                          STH H DA,9C        STORE SNS BYTE 2
1772    2EE3    FILT 190                                  H0=0$K0E           SET CE+DE+UC FOR STATUS
1774    3C25    FILT 191                                  P0=P0$K20          SET CONTINGENT CONNECTION BIT
1776    D8FA    FILT 192                   194  STR9A     BR IF FGA5=0       BR IF NO UNIT EXCEPTION
1778    3E13    FILT 193  SETUEX                          H0=H0$K01          SET UNIT EXCEPTION IN STATUS
177A    7C52    FILT 194  STR9A                           STH P DA,9A        STORE SNS BYTES 0&1
177C    2F07    FILT 195                                  H1=0               ZERO CHAN STATUS
177E    FD82    FILT 196  TSTSM            198  TSTPCI    BR IF FFI3=0       BR IF MO STATUS MOD
1780    3E45    FILT 197                                  H0=H0$K40          SET STAT MOD STATUS
1782    C986    FILT 198  TSTPCI           200  RSTTRP    BR IF FFI4=0       BR IF NOT PCI
1784    3F85    FILT 199                                  H1=H1$K80          SET PCI IN STATUS
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS

1786    1D04    FILT 200    RSTTRP                           RST FIA K=A0       RESET PCI AND TRAP LATCHES
1788    C891    FILT 201                  205     SETICL     BR IF FGA4=1       BR IF INCORRECT LENGTH
178A    F993    FILT 202                  206     ZERT       BR IF FFI7=1       BR IF SLI
178C    4886    FILT 203                                     I=I                TEST FOR CCW COUNT 0
178E    C493    FILT 204                  206     ZERT       BR IF Z=0          BR IF LENGTH OK
1790    3F45    FILT 205    SETICL                           H1=H1$K40          SET ICL FOR STATUS
1792    2A07    FILT 206    ZERT                             T0=0
1794    0F85    FILT 207                                     Z=H1*-K80          TEST FOR CHNL STATUS.ZAP PCI BIT
1796    C49A    FILT 208                  210     SETAUX     BR IF ZNZ          BR IF HAD ERROR
1798    EA24    FILT 209                  215     PASS       BR IF H0 BIT6=0    BR IF NO UNIT CHECK
179A    2843    FILT 210    SETAUX                           T1=0$K04              SET AUX ADDR 0014 T/
179C    3B15    FILT 211                                     T1=T1$K10             ADDR CE AID BUFFER
179E    5C42    FILT 212                                     RDH P DA,98        GET FAILING CCW ADDR+8
17A0    7EA8    FILT 213                                     STH H AS,T+2       STORE ERROR STATUS IN AUX 0014
17A2    7CA0    FILT 214                                     STH P AS,T+0       STORE ERROR CCW ADDR IN AUX 0016
17A4    92BE    FILT 215     PASS       FILX 441    CHNEND     BR
                FILT 216    AEND
```

****
* CROSS REFERENCE FOR CSECT FILT *
****

```
FILT 025    FILT 018  FILX 140
FILT 030    FILT 027  FILX 096  FILX 132
FILT 038    FILF 244
FILT 042    FILT 034
FILT 045    FILT 042
FILT 057    FILT 053  FILT 054
FILT 061    FILT 057
FILT 067    FILT 066  FILT 071  FILT 074  FILT 075
FILT 074    FILT 070
FILT 076    FILT 068  FILT 069
FILT 078    FILT 037
FILT 079    FILT 032  FILT 039  FILT 041  FILT 044
FILT 081    FILT 049  FILT 051
FILT 086    FILT 025  FILT 026
FILT 091    BMCK 067
FILT 092    FILT 089
FILT 093    BWRP 053
FILT 094    FILT 092
FILT 095    BWRP 018
FILT 096    FILT 094
FILT 097    FILT 100
FILT 099    FILT 088
FILT 112    FILT 110
FILT 113    FILT 113  FILX 356
FILT 115    FILX 475
FILT 123    FILT 123
FILT 130    FILT 130
FILT 131    FILT 112
FILT 132    FILT 101
FILT 136    FILT 132  FILT 133
FILT 145    FILT 143
```

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT FILT *
                                  **********************************
FILT 146     FILT 170
FILT 148     FILT 139
FILT 157     FILT 149
FILT 158     FILT 150
FILT 159     FILT 155
FILT 163     FILT 111
FILT 165     FILT 137  FILT 140  FILT 163
FILT 177     FILX 007
FILT 179     FILT 169
FILT 181     FILT 172
FILT 185     FILT 183
FILT 187     FILT 185
FILT 188     FILT 147  FILT 162  FILT 167
FILT 189     FILT 180
FILT 193     FILT 175
FILT 194     FILT 176  FILT 192
FILT 196     FILT 098
FILT 198     FILT 196
FILT 200     FILT 198
FILT 205     FILT 201
FILT 206     FILT 202  FILT 204
FILT 210     FILT 208
FILT 215     FILT 209
```

FILX DESCRIPTIVE TEXT

ENTRY POINTS

DECODE N — THIS ENTRY IS FROM THE FILT ROUTINE TO DECODE THE COMMAND. A SIXTEEN WAY BRANCH MAKES THE INITIAL COMMAND SELECTION ON THE FOUR LOW ORDER BITS. FINAL SELECTION AND VERIFICATION OF EACH COMMAND IS MADE BEFORE PROCEEDING TO THE EXECUTION SEQUENCE.

NTORST — THIS ENTRY PROVIDES A RESET OF THE NTO LATCH TO ALLOW ORIENTING THE FILE FOR THE NEXT COMMAND. IT IS USED FOLLOWING THE OE TESTING FOR A CONTROL COMMAND.

INVSEQ — THIS ENTRY SETS THE INVALID SEQUENCE AND COMMAND REJECT SENSE BITS. IT IS USED FROM VARIOUS POINTS THAT DETECT COMMAND SEQUENCE ERRORS.

GET8E — THIS IS AN ENTRY USED TO END THE MICRO-ROUTINE FOR THE COMMAND. IF THE OPERATION ORIGINALLY ENTERED THE ROUTINE DURING THE INITIAL SEQUENCE, THE ACTIVE BIT IS SET AND THE OPERATION RETURNED TO ICYCLES. WHEN THE ROUTINE IS ENTERED THROUGH THE TRAP, THE OPERATION ENDS WITH RETURN FROM TRAP.

TRPRTN — THIS ENTRY IS USED TO RESTART A RECALIBRATE OR SEEK OPERATION THAT WAS STOPPED BECAUSE THE FILE WAS STILL ERASING. THE EXIT WAS MADE FROM THE ROUTINE WITHOUT RESETTING THE COMMAND DATA.

PROTEC — THIS ENTRY IS USED TO SET THE CONTINGENT CONNECTION INDICATOR AND THE FILE PROTECT SENSE BIT BEFORE SETTING THE SENSE BYTES TO END THE OPERATION.

GETCYL — THIS ENTRY IS USED TO ENTER THE SEEK ROUTINE FOR A HEAD ADVANCE DURING A MULTIPLE TRACK READ OR SEARCH OPERATION.

SETCYL — THIS ENTRY IS USED TO REINITIATE THE READ OR SEARCH OPERATION AFTER TESTING FOR A SUCCESSFUL HEAD COMPARE FOLLOWING A MULTIPLE TRACK HEAD ADVANCE.

ZROSN2 — THIS IS AN ENTRY INTO THE CHAIN END SEQUENCE ROUTINE THAT FIRST RESETS SENSE BYTE 2 BEFORE ENTERING THE NORMAL ENDING SEQUENCE.

CHNEND — THIS IS THE NORMAL CHAIN END SEQUENCE ENTRY USED TO SET THE INTERRUPT BUFFER AND THE INTERRUPT LATCH. THE RETURN FROM TRAP OCCURS AFTER RESETTING THE FILE CONTROL OP AND FLAG REGISTERS.

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

            FILX 001   T                          2311 FEATURE   COMMAND VERIFICATION AND EXECUTION
            FILX 002   *          ***************************************************************
            FILX 003   *          *                                                             **
            FILX 004   *          *                  FILE OP DECODE - COMMON                    **
            FILX 005   *          *                                                             **
            FILX 006   *          ***************************************************************
0680  9758  FILX 007    DECODE 0  FILT 177  SELSTA     BR                     OP REG FAILURE SET EQUIP CHK
0682  8FFB  FILX 008    DECODE 1       197  ERASE  N   N=FOP BITS23          ERASE OR SEARCH ID
0684  EFB8  FILX 009    DECODE 2       052  IPLRCT     BR IF FOP2=0           READ IPL OR READ COUNT
0686  EFE2  FILX 010    DECODE 3       078  CTLOP      BR IF FOP2=0           NO OP OR RECALIBRATE
0688  9874  FILX 011    DECODE 4       225  SENSE      BR                     SENSE OP
068A  8F9B  FILX 012    DECODE 5       149  WRDARO N   N=FOP BITS23          WRITE DATA OR RO
068C  EFCC  FILX 013    DECODE 6       062  RDDARO     BR IF FOP2=0           READ DATA OR RO
068E  EFE2  FILX 014    DECODE 7       078  CTLOP      BR IF FOP2=0           SEEK OR RESTORE
```

```
ADDR    WORD  SEQUENCE NO.  LABEL      NEXTSEQ  NEXTLABEL  STATEMENT              COMMENTS
0690    85AC    FILX 015    DECODE 8       171  INVCMN     BR                     INVALID CMND
0692    8FBB    FILX 016    DECODE 9       165  WRHA    N  N=FOP BITS23         WRITE HA OR SEARCH KEY
0694    EFC8    FILX 017    DECODE A       060  RDHA       BR IF FOP2=0           READ HA
0696    EFE2    FILX 018    DECODE B       078  CTLOP      BR IF FOP2=0           SEEK CYL OR SEEK HD
0698    85AC    FILX 019    DECODE C       171  INVCMN     BR                     INVALID CMND
069A    8FDB    FILX 020    DECODE D       181  WRKD    N  N=FOP BITS23         WRITE KEY &DATA OR SCAN
069C    EFD6    FILX 021    DECODE E       067  RDCKD      BR IF FOP2=C           READ COUNT KEY DATA
069E    EF91    FILX 022    DECODE F       015  DECODE 8   BR IF FOP2=1           SET FILE MASK OR SPACE COUNT
06A0    DF91    FILX 023                   015  DECODE 8   BR IF FOP1=1           BR IF INVALID COMMAND
06A2    5E32    FILX 024                                  RDH H DA,8E            GET STATS
06A4    FFF6    FILX 025                   029  SPACE      BR IF FOP3=0           BR IF SPACE COUNT OP
06A6    3E00    FILX 026                                  SET FIB K=80           SET NON TIME ORIENTED
06A8    CD2C    FILX 027                   041  RMVSIO     BR IF P1 BIT0=0        BR IF SET FILE MASK ALLOWED
06AA    859A    FILX 028    SEQ            162  INVSEQ     BR
06F6    DE2D    FILX 029    SPACE          041  RMVSIO     BR IF H0 BIT1=1        BR IF FIRST OP IN CHAIH
06F8    F5AB    FILX 030                   028  SEQ        BR IF DASI3=1          BR IF ERASE GATE ON
06FA    EC31    FILX 031                   043  PIND       BR IF P0 BIT2=1        BR IF CHND FRM SNS,CTL
06FC    2C33    FILX 032                                  P0=0$K03
06FE    4FCF    FILX 033                                  FOP=P0                 STRIP KEY&DATA BITS FROM OP REG
0700    5DD0    FILX 034                                  P1=P1L                 SAVE SEEK MASK
0702    3DB5    FILX 035                                  P1=P1$KB0              SET MASK TO INH ALL WRITES
0704    5E42    FILX 036                                  RDH H DA,98            GET CCW ADDR
0706    3E00    FILX 037                                  SET FIB K=80           SET NOT TIME ORIENTED
0708    0E04    FILX 038                                  RST FIB K=20           ISSUE COLD START RESET
070A    2C45    FILX 039                                  P0=0$K40               SET SCAN COUNT PREVIOUS OP
070C    85F2    FILX 040                   206  DATAOP     BR                     BR TO GIVE GO PULSE
06AC    1E45    FILX 041    RMVSIO                        H0=H0*-K40             STRIP CAME FROM SIO STAT
06AE    7E32    FILX 042                                  STH H DA,8E            STORE STATS
06B0    2C07    FILX 043    PIND                          P0=0                   SET PREVIOUS OP
06B2    FFE7    FILX 044                   089  SETFM      BR IF FOP3=1           BR IF SET FILE MASK OP
06B4    F8B4    FILX 045    SCINDX         045  SCINDX     BR IF FGA7=0           WAIT FOR INDEX
06B6    8632    FILX 046                   115  STRFM      BR                     GO STORE PREV OP AND END
                FILX 047    *              ***************************************************************
                FILX 048    *              *                                                             **
                FILX 049    *              *                IPL OR READ COUNT                            **
                FILX 050    *              *                                                             **
                FILX 051    *              ***************************************************************
06B8    FFDB    FILX 052    IPLRCT         069  RDCTSP     BR IF FOP3=1           BR IF READ COUNT OP
06BA    DF91    FILX 053                   015  DECODE 8   BR IF FOP1=1           BR IF INVALID CMND
06BC    CD11    FILX 054                   015  DECODE 8   BR IF P1 BIT0=1        BR IF PREVIOUS SET FILE MASK IPL
06BE    3E00    FILX 055                                  SET FIB K=80           SET NON TIME ORIENTED
06C0    F5C1    FILX 056    IPLRD          056  IPLRD      BR IF DASI3=1          BR IF ERASE GATE ON
06C2    2C65    FILX 057                                  P0=0$K60               TURN ON CC AND SLI BITS
06C4    4DCF    FILX 058                                  FF0=P0                 SET CNND CHN IN FLAG REG
06C6    872A    FILX 059                   341  RECAL      BR                     GO EXECUTE RECALIBRATE
06C8    FF90    FILX 060    RDHA           015  DECODE 8   BR IF FOP3=0           BR IF INVALID CMND
06CA    85F0    FILX 061    SETPO          205  RSPOGO     BR
06CC    E851    FILX 062    RDDARO         064  SMTST      BR IF P0 BIT6=1        BR IF PREVIOUS SEARCH
06CE    F84A    FILX 063    RDCKDA         061  SETPO      BR IF P0 BIT7=0        BR IF NO PREVIOUS SEARCH
06D0    FFCB    FILX 064    SMTST          061  SETPO      BR IF FOP3=1           BR IF READ RO OR READ CKD
06D2    D1CA    FILX 065                   061  SETPO      BR IF DASI5=0          BR IF NO STATUS MODIFIER
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

06D4    8594    FILX 066                159  SET04      BR
06D6    FFCE    FILX 067    RDCKD       063  RDCKDA     BR IF FOP3=0         BR IF READ KEY DATA
06D8    DC4E    FILX 068                063  RDCKDA     BR IF P0 BIT1=0      BR IF PREV OP WAS NOT SP COUNT
06DA    DC4A    FILX 069    RDCTSP      061  SETP0      BR IF P0 BIT1=0      BR IF PREV OP WAS NOT SP COUNT
06DC    3E00    FILX 070                             SET FIB K=80           SET NON TIME ORIENTED
06DE    0EC4    FILX 071                             RST FIB K=20           COLD START RESET
06E0    85F0    FILX 072                205  RSPOGO     BR
                FILX 073    *          ***************************************************************
                FILX 074    *          *                                                             **
                FILX 075    *          *                CONTROL OPS (NTO RESET)                      **
                FILX 076    *          *                                                             **
                FILX 077    *          ***************************************************************
06E2    DF91    FILX 078     CTLOP       015  DECODE 8  BR IF FOP1=1         BR IF INVALID CMND
06E4    9058    FILX 079                 080  ABCDE      BR
1058    3E00    FILX 080     ABCDE                   SET FIB K=80           SET NON TIME ORIENTED LATCH
105A    2C25    FILX 081                             P0=0$K20               SET PREV OP IND
105C    7C52    FILX 082                             STH P DA,9A            STORE PREV OP
105E    8714    FILX 083                 324  CTLS       BR
                FILX 084    *          ***************************************************************
                FILX 085    *          *                                                             **
                FILX 086    *          *                 SET FILE MASK                               **
                FILX 087    *          *                                                             **
                FILX 088    *          ***************************************************************
06E6    5DB8    FILX 089     SETFM                   RDB P1 T+1             GET FILE MASK BYTE
06E8    5886    FILX 090                             I=I-1                  DECREMENT CCW COUNT
06EA    CDEF    FILX 091                 093  DCIDEC     BR IF FFI0=1         BR IF DATA CHAIN
06EC    860E    FILX 092    IGNRDC       097  FMRETN     BR                       CONTINUE WITH SET FILE MASK
06EE    C4EC    FILX 093    DCIDEC       092  IGNRDC     BR IF ZNZ            BR IF COUNT NOT ZERO
06F0    7C52    FILX 094                             STH P DA,9A            SAVE FILE MASK BYTE
06F2    5E42    FILX 095    CCWADR                   RDH H DA,98            RESTORE CCW ADDRESS IN H
06F4    830A    FILX 096                FILT 030   FRSTBT     BR                       GET NEXT CCW FOR DATA CHAIN
060E    0D83    FILX 097    FMRETN                   Z=P1*-K08              TEST FOR VALID MASK LO
0610    F094    FILX 098                 100  INVFM      BR IF LZNZ           BR IF MASK INVALID
0612    ED16    FILX 099                 101  NONSNS     BR IF P1 BIT2=0      BR IF VALID MASK HI
0614    98A6    FILX 100    INVFM        300  CHNSET     BR
0616    3D13    FILX 101    NONSNS                   P1=P1$K01              SET NONSENSE STORED BIT
0618    2C25    FILX 102                             P0=0$K20               SET PREVIOUS OP
061A    C920    FILX 103                 106  TSTB3      BR IF P1 BIT4=0      BR IF INH ALL$ALLOW ALL SEEKS
061C    FD26    FILX 104                 109  TSTB0      BR IF P1 BIT3=0      MASK ALLOWS ALL SEEKS TST WRITES
061E    3DF3    FILX 105                             P1=P1$K0F              SET ALG TO INH ALL SEEKS
0620    FD26    FILX 106    TSTB3        109  TSTB0      BR IF P1 BIT3=0      BR IF  ALLOW ALL SEEKS
0622    3DD3    FILX 107                             P1=P1$K0D              SET ALG TO INH SEEK&SEEK CYL
0624    1D15    FILX 108                             P1=P1*-K10             REMOVE BIT 3 OF ALGORITHM
0626    CD2F    FILX 109    TSTB0        113  TSTB1      BR IF P1 BIT0=1      BR IF ALLOW ALL$INH R0-HA-CKD
0628    3D85    FILX 110                             P1=P1$K80              SET ALG TO INH SET FILE MASK
062A    DD32    FILX 111                 115  STRFM      BR IF P1 BIT1=0      BR IF ALGORITHM COMPLETE
062C    1DFD    FILX 112                             P1=P1<KF0              SETS ALG TO INHIBIT ALL WRITES
062E    DD33    FILX 113    TSTB1        115  STRFM      BR IF P1 BIT1=1      BR IF ALG IS COMPLETE
0630    3DA5    FILX 114                             P1=P1$KA0              SET ALG TO INH WR HA-R0-CKD
0632    7C52    FILX 115    STRFM                    STH P DA,9A            STORE FILE MASK AND PREV OP ALGS
0634    2F45    FILX 116    COMEND                   H1=0$K40               PRESET INCORRECT LENGTH BIT
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS
0636    CDDB    FILX 117                  142      STASET     BR IF FFI0=1       BR IF DATA CHAIN
0638    F9BF    FILX 118                  121      CORLEN     BR IF FFI7=1       BR IF SLI ON
063A    4886    FILX 119                             I=I                         TEST FOR CCW COUNT 0
063C    C4DA    FILX 120                  142      STASET     BR IF ZNZ          BR IF INCORRECT LEGNTH
063E    DDD8    FILX 121    CORLEN        141      CEADE      BR IF FFI1=0       BR IF NOT COMMAND CHAIN
0640    2E08    FILX 122    NTORST                             SET FIB K=40       SER FLAG BITS TO ZERO
0642    0E04    FILX 123    CLDRST                             RST FIB K=20       COLD START RESET
0644    5E42    FILX 124                                       RDH H DA,98        GET CCW ADDR
0646    5CF0    FILX 125                                       RDH P H            READ NEXT OP CODE
0648    0CC3    FILX 126                                       Z=P0*-K0C          CHECK FOR SENSE OR TIC
064A    C4D5    FILX 127                  132      NXTCCW     BR IF Z=0          BR IF SENSE OR TIC
064C    0C3B    FILX 128                                       Z=P0<K03           CHECK FOR RECAL OR NO-OP
064E    F0DE    FILX 129                  133      NOTREC     BR IF LZNZ         BR IF NOT RECAL OR NO-OP
0650    DEDE    FILX 130                  133      NOTREC     BR IF DS1=0        BR IF OFF LINE
0652    EEDF    FILX 131                  133      NOTREC     BR IF DS2=1        BR IF UNSAFE
0654    830A    FILX 132    NXTCCW  FILT 030       FRSTBT     BR                 GET NEW CCW
065E    EDD7    FILX 133    NOTREC        140      ADROK      BR IF FFI BIT2=1   BR IF NO ERROR CONDITION
0660    2F6B    FILX 134                                       H1=H1+K06          CURRENT CCW + 6
0662    58F8    FILX 135                                       RDH I H+2          LOAD NEXT CCW COUNT FIELD INTO I
0664    7E42    FILX 136                                       STH H DA,98        STORE UPDATED CCW ADR
0666    2F07    FILX 137                                       H1=0               ZERO SENSE BYTE 1
0668    2F65    FILX 138                                       H0=0$K60           SET INTV REQ & CONTINGENT CONN
066A    85BA    FILX 139                  178      SETSEN     BR                 GO STORE SENSE
0656    8D87    FILX 140    ADROK   FILT 025       TRAP    N  N=FFI BITS23       GO CHECK IF UNUSUAL CONDITION
0658    2F07    FILX 141    CEADE                              H1=0               ZERO CHNL STATUS
065A    2EC3    FILX 142    STASET                             H0=0$K0C           SET CHAN END AND DEVICE END
065C    92BA    FILX 143                  439      ZROSN2     BR
                FILX 144    *          ********************************************************************************
                FILX 145    *          *                                                                              **
                FILX 146    *          *                    WRITE DATA OR RO (05 OR 15)                               **
                FILX 147    *          *                                                                              **
                FILX 148    *          ********************************************************************************
0580    DF88    FILX 149     WRDARO 0      153      WRDATA     BR IF FOP1=0       BR IF WRITE DATA OP
0582    DF8E    FILX 150     WRDARO 1      156      WRROOP     BR IF FOP1=0       BR IF WRITE RO OP
0584    85AC    FILX 151     WRDARO 2      171      INVCMN     BR                 INVALID COMMAND DECODED
0586    85AC    FILX 152     WRDARO 3      171      INVCMN     BR                 INVALID COMMAND DECODED
0588    2D10    FILX 153     WRDATA                             SET FIA K=01       SET READ CLOCK CONTROL
058A    E833    FILX 154                   174      DASI5      BR IF P0 BIT6=1    BR IF PREVIOUS SEARCH ID OR KEY
058C    859A    FILX 155                   162      INVSEQ     BR                 INVALID SEQUENCE
058E    C819    FILX 156     WRROOP        161      TDASI      BR IF P0 BIT4=1    BR IF PREVIOUS SEARCH HA
0590    FC1A    FILX 157                   162      INVSEQ     BR IF P0 BIT3=0    BR IF NOT PREV WRITE HA
0592    DD36    FILX 158     P1TEST        176      FIPROT     BR IF P1 BIT1=0    BR IF FILE PROTECTED
0594    2C43    FILX 159     SET04                              P0=0$K04           SET PREVIOUS OP
0596    85F2    FILX 160                   206      DATAOP     BR
0598    D193    FILX 161     TDASI         158      P1TEST     BR IF DASI5=1      BR IF STATUS MOD&NOT SHORT SCH
059A    2F15    FILX 162     INVSEQ                             H1=0$K10           SET INVALID SEQ BIT FOR SENSE
059C    85B8    FILX 163                   177      COMREJ     BR
059E    DFD8    FILX 164     WRHAOP        193      FLGRST     BR IF FOP1=0       BR IF NOT INVALID COMMAND
05A0    85AC    FILX 165     WRHA    0     171      INVCMN     BR
05A2    859E    FILX 166     WRHA    1     164      WRHAOP     BR                 WRITE HOME ADDRESS
05A4    2C23    FILX 167     WRHA    2                         P0=0$K02           PRESET SEARCH KEY ALGORITHM
```

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
05A6  FFEE   FILX 168     WRHA   3     204   TSTOP      BR IF FOP3=0       BR IF SEARCH KEY OP
05A8  2C83   FILX 169                                    P0=0$K08           PRESET SEARCH HA ALGORITHM
05AA  CFF2   FILX 170                 206   DATAOP     BR IF FOP0=0       BR IF SCH HA EQUAL
05AC  2F07   FILX 171     INVCMN                       H1=0               ZERO SENSE BYTE 1
05AE  85B8   FILX 172                 177   COMREJ     BR
05B0  F81A   FILX 173     WRKDOP      162   INVSEQ     BR IF P0 BIT7=0    BR IF NO PREV SEARCH
05B2  D19A   FILX 174     DASI5       162   INVSEQ      BR IF DASI5=0     BR IF NO STATUS MOD
05B4  FD70   FILX 175                 205   RSPOGO     BR IF P1 BIT3=0    BR IF NO FILE PROTECT
05B6  2F43   FILX 176     FIPROT                       H1=0$K04           SET FILE PROTECT FOR SENSE
05B8  2EA5   FILX 177     COMREJ                       H0=0$KA0           SET CMND REJ&CONTINGENT CONN SNS
05BA  7E52   FILX 178     SETSEN                       STH H DA,9A        STORE SENSE BYTES 0&1
05BC  2E23   FILX 179                                  H0=0$K02           SET UNIT CHECK FOR STATUS
05BE  92B8   FILX 180                 438   ZROSTA     BR
05C0  DFB0   FILX 181     WRKD   0    173   WRKDOP     BR IF FOP1=0       BR IF WRITE KEY DATA OP
05C2  DFC8   FILX 182     WRKD   1    185   WRCKD      BR IF FOP1=0       BR IF WRITE COUNT KEY DATA OP
05C4  EFF1   FILX 183     WRKD   2    205   RSPOGO     BR IF FOP2=1       BR IF SCAN OP
05C6  85AC   FILX 184     WRKD   3    171   INVCMN     BR
05C8  E89B   FILX 185     WRCKD       162   INVSEQ     BR IF FGA6=1       BR IF ON DEFECTIVE TRACK
05CA  E853   FILX 186                 190   SETCKD     BR IF P0 BIT6=1    BR IF PREV SCH ID OR KEY
05CC  D81A   FILX 187                 162   INVSEQ     BR IF P0 BIT5=0    BR IF NOT PREV WRITE CKD
05CE  ED14   FILX 188     P2TEST      159   SET04      BR IF P1 BIT2=0    BR IF NOT FILE PROTECT
05D0  85B6   FILX 189                 176   FIPROT     BR
05D2  2D10   FILX 190     SETCKD                       SET FIA K=01       SET READ CLOCK CONTROL
05D4  D1CF   FILX 191                 188   P2TEST     BR IF DASI5=1      BR IF STATUS MOD&NOT SHORT SCH
05D6  859A   FILX 192                 162   INVSEQ     BR
05D8  2E08   FILX 193     FLGRST                       SET FIB K=40       RESET FLAG BITS 6,7
05DA  2C15   FILX 194                                  P0=0$K10           SET WRITE HA ALGORITHM
05DC  DD73   FILX 195                 206   DATAOP     BR IF P1 BIT1=1    BR IF NOT FILE PROTECT
05DE  85B6   FILX 196                 176   FIPROT     BR
05E0  85AC   FILX 197     ERASE  0    171   INVCMN     BR                 INVALID COMMAND***WRITE SPECIAL
05E2  DFC8   FILX 198     ERASE  1    185   WRCKD      BR IF FOP1=0       BR IF WRITE COUNT KEY DATA OP
05E4  85AC   FILX 199     ERASE  2    171   INVCMN     BR                 INVALID COMMAND DECODED
05E6  DC6C   FILX 200     ERASE  3    203   OPSPCT     BR IF P0 BIT1=0    BR IF PREV OP WAS NOT SP COUNT
05E8  3E00   FILX 201                                  SET FIB K=80       SET NON TIME ORIENTED
05EA  0E04   FILX 202                                  RST FIB K=20       COLD START RESET
05EC  2C33   FILX 203     OPSPCT                       P0=0$K03           SET SEARCH ID ALGORITHM
05EE  CFF2   FILX 204     TSTOP       206   DATAOP     BR IF FOP0=0       BR IF SEARCH EQUAL
05F0  2C07   FILX 205     RSPOGO                       P0=0               SET PREVIOUS OP ALG
05F2  D6F7   FILX 206     DATAOP      208   MMSK1      BR IF MMSK1=1      BR IF IN FILE TRAP
05F4  221E   FILX 207                                  SET MMSK K=71      TURN MMSK 7 ON TO BLOCK TRAPS
05F6  3D00   FILX 208     MMSK1                        SET FIA K=80       ISSUE GO PULSE
05F8  7C52   FILX 209     STP0                         STH P DA,9A        STORE PREVIOUS OP ALG
05FA  2C15   FILX 210                                  P0=0$K10           SET CONTROL BIT
05FC  4BCF   FILX 211                                  TGR0=P0            RAISE CONTROL TAG
05FE  1D00   FILX 212     GET8E                        RST FIA K=80       RESET TRAP LATCH
0600  5C32   FILX 213     CDEND                        RDH P DA,8E        GET STATS
0602  D686   FILX 214     TRAPBR      216   RTNIC      BR IF MMSK1=0      BR IF NOT IN FILE TRAP
0604  0222   FILX 215                                  RTN V MMSK1=0      RETURN FROM TRAP
0606  2C85   FILX 216     RTNIC                        P0=0$K80           SET ACTIVE STAT
0608  7C32   FILX 217                                  STH P DA,8E        STORE STATS
060A  021E   FILX 218                                  RST MMSK K=71      TURN OFF MMSK 7
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS

060C  810C    FILX 219              CCOM 155  CCOB      BR                     SET CCO
              FILX 220    *            ***************************************************************
              FILX 221    *            *                                                             **
              FILX 222    *            *                SENSE & SEEK-SENSE COMMON                    **
              FILX 223    *            *                                                             **
              FILX 224    *            ***************************************************************
1874  3E00    FILX 225     SENSE                    SET FIB K=80          SET NON TIME ORIENTED
1876  5FFF    FILX 226                              H1=FOP                GET OP
1878  F0DD    FILX 227                229  SNSEX     BR IF HZ=0            BR IF VALID OP
187A  85AC    FILX 228                171  INVCMN    BR
185C  1C25    FILX 229     SNSEX                    P0=PC*-K20            STRIP CONTINGENT CONNECTION BIT
185E  7C52    FILX 230                              STH P DA,9A           SAVE ALTERED SENSE
1860  5E32    FILX 231     SKSNS                    RDH H DA,8E           GET STATS
1862  1E45    FILX 232                              H0=H0*-K40            REMOVE CAME FROM SIO STAT
1864  7E32    FILX 233                              STH H DA,8E           SAVE STATS
1866  2E07    FILX 234                              H0=0                        SET
1868  2F25    FILX 235                              H1=0$K20                AUX ADDRESS
186A  3F43    FILX 236                              H1=H1$K04                 0024
186C  FBAA    FILX 237                241  SNSRD     BR IF FOP7=0          BR IF SENSE OP
186E  5DB8    FILX 238     SKRD                     RDB P1 T+1            GET A BYTE OF SEEK DATA
1870  7DE8    FILX 239                              STB P1 AS,H+1         SAVE BYTE IN AUX
1872  984E    FILX 240                259  SKIPON    BR
182A  F930    FILX 241     SNSRD       244  STR01     BR IF P1 BIT7=0       BR IF SENSE IS STORED
182C  4C52    FILX 242                              RDH P DC,9A                ZERO SENSE
182E  7C62    FILX 243                              STH P DA,9C                     BYTES
1830  7CE8    FILX 244     STR01                    STH P AS,H+2          STORE SNS BYTES 0&1 IN AUX
1832  5C62    FILX 245                              RDH P DA,9C           GET SNS BYTE 2
1834  5EDF    FILX 246                              P1=DS                 SET SENSE BYTE 3
1836  1DA3    FILX 247                              P1=P1*-K0A            STRIP BITS 4,6
1838  DD3C    FILX 248                250  DUMPSN    BR IF P1 BIT1=0       BR IF NOT ON LINE
183A  3D83    FILX 249                              P1=P1$K08             OR IN ON LINE BIT
183C  7CE0    FILX 250     DUMPSN                   STH P AS,H+0
183E  4C52    FILX 251                              RDH P DC,9A           ZERO SENSE BYTES 4 AND 5
1840  3F15    FILX 252                              H1=H1$K10             SET AUX ADDR 0034
1842  7CEA    FILX 253                              STH P AS,H-2          STORE BYTES 4&5 IN AUX
1844  1F1D    FILX 254     ADRINK                   H1=H1<K10             SET AUX ADDR FOR NEXT BYTE
1846  FBEF    FILX 255     TSTFOP      238  SKRD      BR IF FOP7=1          BR IF SEEK OP
1848  5DE8    FILX 256                              RDB P1 AS,H+1         GET AS SENSE BYTE
184A  D5CF    FILX 257                259  SKIPON    BR IF DASI1=1         BR IF SKIP IS ON
184C  7DB8    FILX 258                              STB P1 T+1            STORE A SENSE BYTE
184E  5886    FILX 259     SKIPON                   I=I-1                 DECREMENT CCW COUNT
1850  C4FD    FILX 260                267  TSTCTR    BR IF Z=0             BR IF COUNT AT ZERO
1852  CB46    FILX 261     LASTST      255  TSTFOP    BR IF H1 BIT4=0       BR IF NOT LAST BYTE
1854  1FEB    FILX 262                              H1=H1<K0E             RESTORE LO AUX ADDR
1856  FF44    FILX 263                254  ADRINK    BR IF H1 BIT3=0       BR IF NOT LAST BYTE
1858  FBA9    FILX 264                266  SKX       BR IF FOP7=1          BR IF SEEK
185A  8634    FILX 265                116  COMEND    BR
1828  988E    FILX 266     SKX         288  SEEKOP    BR
187C  CDA1    FILX 267     TSTCTR      276  SETADR    BR IF FFIO=1          BR IF DATA CHAIN
187E  CB06    FILX 268                272  SHORT     BR IF H1 BIT4=0       BR IF NOT LAST BYTE
1880  FF06    FILX 269                272  SHORT     BR IF H1 BIT3=0       BR IF NOT LAST BYTE
```

```
ADDR   WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
1882   FB8F    FILX 270                288    SEEKOP     BR IF FOP7=1       BR IF SEEK
1884   8634    FILX 271    SLIISN      116    COMEND     BR
1886   2F45    FILX 272    SHORT                          H1=0$K40           PRESETN INCORECT LENGNTH
1888   FBC7    FILX 273                280    SKERR      BR IF FOP7=1       BR IF SEEK OP
188A   F985    FILX 274                271    SLIISN     BR IF FFI7=1       BR IF SLI
188C   865A    FILX 275                142    STASET     BR
1820   5C84    FILX 276    SETADR                         P=I+1              SET AUX
1822   2D3F    FILX 277                                   P1=P1+K33              ADDRESS 0034
1824   7EC0    FILX 278                                   STH H AS,P+0       STORE NEXT BYTE ADDR FOR DATA CH
1826   86F2    FILX 279                095    CCWADR     BR
18C6   2C13    FILX 280    SKERR                          P0=0$K01           SET SEEK CHECK FOR SNS
18C8   F9CC    FILX 281                283    CMREJ      BR IF FFI7=0       BR IF NOT SLI
18CA   2F07    FILX 282    CHNZER                         H1=0               ZERO CHAN STATUS
18CC   3CA5    FILX 283    CMREJ                          P0=P0$KA0          SET CMND REJECT&CONTIN CONN BITS
18CE   2D07    FILX 284                                   P1=0               ZERO SNS BYTE 1
18D0   7C52    FILX 285                                   STH P DA,9A        STORE SNS
18D2   2EE3    FILX 286                                   H0=0$K0E           SET CE&DE&UC
18D4   92BA    FILX 287                439    ZROSN2     BR
188E   2A07    FILX 288    SEEKOP                         T0=0                       SET AUX
1890   2B25    FILX 289                                   T1=0$K20                   ADDRESS
1892   3B43    FILX 290                                   T1=T1$K04                   0024
1894   5CA8    FILX 291                                   RDH P AS,T+2       GET BYTES 0&1 OF SEEK DATA
1896   4CC6    FILX 292                                   P=P
1898   C4A4    FILX 293                299    SKCHK      BR IF ZNZ          BR IF BYTES ARE NON ZERO
189A   5CA0    FILX 294                                   RDH P AS,T+0       GET BYTES 2&3 OF SEEK DATA
189C   3817    FILX 295                                   T1=T1$K11          SET AUX ADR 0037
189E   5EAA    FILX 296                                   RDH H AS,T-2       GET BYTES 4&5 OF SEEK DATA
18A0   6EC5    FILX 297                                   H0=H0$P0           MASK BYTES 2&4 FOR ZERO
18A2   C4AF    FILX 298                307    SAVCYL     BR IF Z=0          BR IF BYTES 2&4 ARE VALID
18A4   2C13    FILX 299    SKCHK                          P0=0$K01           SET SEEK CHECK FOR SENSE
18A6   2F45    FILX 300    CHNSET                         H1=0$K40           SET WLR
18A8   CDC5    FILX 301                306    ILON       BR IF FFI0=1       BR IF DATA CHAIN
18AA   F9C0    FILX 302                304    TSTI       BR IF FFI7=0       BR IF NOT SLI
18AC   98CA    FILX 303    NOIL        282    CHNZER     BR
18C0   4886    FILX 304    TSTI                           I=I                MASK FOR CCW COUNT ZERO
18C2   C4AD    FILX 305                303    NOIL       BR IF Z=0          BR IF COUNT ZERO
18C4   98CC    FILX 306    ILON        283    CMREJ      BR
18AE   5DE9    FILX 307    SAVCYL                         H0=P1              SAVE CYL NO
18B0   6BE3    FILX 308                                   T1=T1+H0           MASK FOR VALID CYL NO
18B2   F4B8    FILX 309                312    TSTHD      BR IF AC=0         BR IF VALID CYL
18B4   0FFF    FILX 310                                   Z=H0<KFF           MASK FOR CYL 255
18B6   C4A4    FILX 311                299    SKCHK      BR IF ZNZ          BR IF INVALID CYL
18B8   2CA3    FILX 312    TSTHD                          P0=0$K0A           SET CONSTANT
18BA   7CF3    FILX 313                                   P0=P0-H1           MASK FOR VALID HD
18BC   F4A4    FILX 314                299    SKCHK      BR IF AC=0         BR IF INVALID HD
18BE   924E    FILX 315                385    FWDSET     BR
               FILX 316    *          ********************************************************************
               FILX 317    *          *                                                                  **
               FILX 318    *          *              RECALIBRATE OR SEEK, AFTER ERASE IS FINISHED        **
               FILX 319    *          *                                                                  **
               FILX 320    *          ********************************************************************
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS
070E    1D00    FILX 321    TRPRTN                         RST FIA K=80       RESET TRAP LATCH
0710    E976    FILX 322                  330     CETIME   BR IF P1 BIT6=0    BR IF CHANNEL END TIME FOE RECAL
0712    5C52    FILX 323                                   RDH P DA,9A        GET FILE MASK ALGORITHM
0714    F5A0    FILX 324    CTLS          336     EXCTLS   BR IF DASI3=0      BR IF NOT ERASE GATE
0716    2D23    FILX 325                                   P1=0$K02           SET TRAP GATE BIT
0718    3D15    FILX 326                                   P1=P1$K10          SET CONTROL BIT
071A    1D00    FILX 327    RSTWD                          RST FIA K=80       RESET TRAP LATCH
071C    48DF    FILX 328                                   TGRO=P1            SET TRAP GATE
071E    0222    FILX 329    RTNWD                          RTN V MMSK1=0      RETURN FROM TRAP
0776    CDFB    FILX 330    CETIME        332     WLRSET   BR IF FFI0=1      BR IF DATA CHAIN
0778    F9FD    FILX 331                  333     NOWLR    BR IF FFI7=1      BR IF SLI
077A    92A4    FILX 332    WLRSET        428     ICL      BR
077C    DD9F    FILX 333    NOWLR         329     RTNWD    BR IF FFI1=1      BR IF COMMAND CHAIN
077E    2E83    FILX 334                                   HO=0$K08           SET CHANNEL END FOR STATUS
0780    92B8    FILX 335                  438     ZROSTA   BR
0720    FFF1    FILX 336    EXCTLS        357     TSTOP4   BR IF FOP3=1      BR IF RECAL RESTORE OR SEEK HD
0722    CBE9    FILX 337                  360     P5TST    BR IF FOP4=1      BR IF SEEK CYL
0724    DBCC    FILX 338                  365     NOOP     BR IF FOP5=0      BR IF NO OP
0726    C96B    FILX 339    P4TST         361     PROTEC   BR IF P1 BIT4=1   FULL SEEK OR RECAL-TST FILE PROT
0728    FFF4    FILX 340                  359     SKEX     BR IF FOP3=0      BR IF FULL SEEK
072A    2E08    FILX 341    RECAL                          SET FIB K=40       RESET FLAG BITS 6,7
072C    2D07    FILX 342                                   P1=0
072E    48DF    FILX 343                                   TGRO=P1            ZERO TAG REG
0730    2D23    FILX 344                                   P1=0$K02           SET RTN TO 000 BIT
0732    4EDF    FILX 345                                   FBO=P1             SET RTN TO 000 ON BUSS
0734    2D1B    FILX 346    TMREC                          P1=P1+K01          ADD TO GET TIME OUT
0736    F0B4    FILX 347                  346     TMREC    BR IF LZNZ         BR IF NOT TIMED OUT
0738    48DF    FILX 348                                   TGRO=P1            SET CONTROL TAG
073A    5E32    FILX 349                                   RDH H DA,8E        GET STATS
073C    1D00    FILX 350                                   RST FIA K=80       RESET TRAP LATCH
073E    D69F    FILX 351                  329     RTNWD    BR IF MMSK1=1      BR IF      IN FILE TRAP
0740    2E85    FILX 352                                   HO=0$K80           SET ACTIVE STAT
0742    7E32    FILX 353                                   STH H DA,8E        STORE STATS
0744    C3C9    FILX 354                  356     IPLOP    BR IF BA4=1       BR IF IPL LATCH ON
0746    810C    FILX 355                CCOM 155     CCOB     BR                    SET CCO
0748    96D8    FILX 356    IPLOP       FILT 113     IPL      BR                 GO TO IPL WAIT LOOP
0770    CBCA    FILX 357    TSTOP4        364     TSTOP5   BR IF FOP4=0      BR IF RECAL OR RESTORE
0772    E96B    FILX 358                  361     PROTEC   BR IF P1 BIT6=1   SEEK HD TEST FOR FILE PROTECT
0774    9860    FILX 359    SKEX          231     SKSNS    BR
0768    D974    FILX 360    P5TST         359     SKEX     BR IF P1 BIT5=0   SEEK CYL TEST FOR FILE PROTECT
076A    2E25    FILX 361    PROTEC                         HO=0$K20           SET CONTINGENT CONNECTION BIT
076C    2F43    FILX 362                                   H1=0$K04           SET FILE PROTECT BIT FOR SNS
076E    85BA    FILX 363                  178     SETSEN   BR
074A    DBA6    FILX 364    TSTOP5        339     P4TST    BR IF FOP5=0      BR IF RECALIBRATE
074C    5E32    FILX 365    NOOP                           RDH H DA,8E        GET STATS
074E    1E45    FILX 366                                   HO=HO*-K40         STRIP SIO IND
0750    FFE0    FILX 367                  373     ISNOOP   BR IF FOP3=0      BR IF NO OP
0752    F9DC    FILX 368    NOPSLI        377     TSTOP3   BR IF FFI7=0      BR IF NOT SLI
0754    CDDD    FILX 369                  377     TSTOP3   BR IF FFI0=1      BR IF DATA CHAIN
0756    DDDC    FILX 370                  377     TSTOP3   BR IF FFI1=0      BR IF NOT CMND CHAIN
0758    7E32    FILX 371    STRNOP                         STH H DA,8E        STORE STATS
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT              COMMENTS

075A    8642    FILX 372                  123   CLDRST     BR                     BR TO GET NEXT CCW
0760    F5D3    FILX 373    ISNOOP        368   NOPSLI     BR IF DASI3=1          BR IF ERASING
0762    CDDF    FILX 374                  378   DCIL       BR IF FFIO=1           BR IF DATA CHAIN
0764    DDD9    FILX 375                  371   STRNOP     BR IF FFI1=1           BR IF COMMAND CHAIN
0766    8658    FILX 376                  141   CEADE      BR
075C    7E32    FILX 377    TSTOP3                         STH H DA,8E            STORE STATS
075E    8634    FILX 378    DCIL          116   COMEND     BR
                FILX 379  *               ***********************************************************
                FILX 380  *               *                                                         **
                FILX 381  *               *                  SEEK  EXECUTE                          **
                FILX 382  *               *                                                         **
                FILX 383  *               ***********************************************************
                FILX 384  ASEQ    ALO7=4E
124E    3F85    FILX 385    FWDSET                         H1=H1$K80              SET FORWARD INDICATOR
1250    7E62    FILX 386                                   STH H DA,9C            SAVE SEEK DATA
1252    FFD6    FILX 387                  389   GETCAR     BR IF FOP3=0           BR IF NOT HEAD SEEK
1254    51EF    FILX 388    GETCYL                         HO=FBI                 SET CYL NUMBER FOR HD SEEK
1256    51CF    FILX 389    GETCAR                         PO=FBI                 GET CYL ADDR TO COMPUTE DIFF
1258    2D07    FILX 390                                   P1=0
125A    4BDF    FILX 391                                   TGRO=P1                RESET TAG REG
125C    2D15    FILX 392                                   P1=0$K10               CONTROL TAG&RST HEAD BIT
125E    7CE3    FILX 393                                   PO=PO-HO               COMPUTE DIFFERENCE
1260    F4E8    FILX 394                  398   RSTHD      BR IF AC=0             BR IF DIR FWD-NO RECOMP NEEDED
1262    2C1B    FILX 395                                   PO=PO+K01              ADD CARRY
1264    1CFF    FILX 396                                   PO=PO<KFF              RECOMPLIMENT
1266    5FFD    FILX 397                                   H1=H1L                 REMOVE FORWARD BIT
1268    4EDF    FILX 398    RSTHD                          FBO=P1                 PUT RESET HD ON BUSS
126A    4BDF    FILX 399                                   TGRO=P1                SET CONTROL TAG
126C    2D07    FILX 400                                   P1=0
126E    4BDF    FILX 401                                   TGRO=P1                RESET TAG REG
1270    4EEF    FILX 402                                   FBO=HO                 PUT CYL NO ON BUSS
1272    2E45    FILX 403                                   HO=0$K40               SET THE SET CYL BIT
1274    4BEF    FILX 404                                   TGRO=HO                PUT SET CYL IN TAG REG
1276    4BDF    FILX 405                                   TGRO=P1                RESET TAG REG
1278    4EFF    FILX 406                                   FBO=H1                 PUT HD NO ON BUSS
127A    2E25    FILX 407                                   HO=0$K20               SET THE SET HD BIT
127C    4BEF    FILX 408                                   TGRO=HO                RAISE SET HD TAG
127E    4BDF    FILX 409                                   TGRO=P1                RESET TAG REG
1280    0CFF    FILX 410                                   Z=PO<KFF               MASK FOR ZERO DIFF
1282    C496    FILX 411                  421   MOTION     BR IF ZNZ              BR IF DIFF NOT ZERO-MOTION SEEK
1284    2F43    FILX 412                                   H1=0$K04               SET SELECT HD BIT
1286    4EFF    FILX 413                                   FBO=H1                 PUT SELECT HD ON BUSS
1288    2EC3    FILX 414                                   HO=0$K0C               SET CE+DE FOR DEVICE STATUS
128A    DFA6    FILX 415                  429   ISSEEK     BR IF FOP1=0           BR IF NOT FROM HD SSITCH ROUTINE
128C    2D15    FILX 416    SETCTL                         P1=0$K10               SET CONTROL BIT
128E    4BDF    FILX 417                                   TGRO=P1                RAISE CONTROL TAG
1290    5E42    FILX 418                                   RDH H DA,98            GET CCW ADDR
1292    1D00    FILX 419                                   RST FIA K=80           RESET TRAP LATCH
1294    0222    FILX 420                                   RTN V MMSK1=0          RETURN FROM TRAP
1296    4ECF    FILX 421    MOTION                         FBO=PO                 PUT DIFFERENCE ON BUSS
1298    2F85    FILX 422                                   H1=0$K80               SETV THE SET DIFF BIT
```

```
ADDR  WORD  SEQUENCE NO.  LABEL    NEXTSEQ  NEXTLABEL  STATEMENT          COMMENTS
129A  4BFF    FILX 423                                  TGRO=H1            RASSE SET DIFF TAG
129C  4BDF    FILX 424                                  TGRO=P1            RESET TAG REG
129E  4EEF    FILX 425                                  FBO=HO             PUT SEEK START ON BUSS
12A0  2D13    FILX 426                                  P1=0$K01           SET CTL PULSE BIT
12A2  4BDF    FILX 427                                  TGRO=P1            ISSUE CONTROL PULSE
12A4  2E83    FILX 428    ICL                           HO=0$K08           SET CHANNEL END FOR STATUS
12A6  2F45    FILX 429    ISSEEK                        H1=0$K40           PRESET INCORRECT LEN STATUS
12A8  CDBB    FILX 430                  439  ZROSN2     BR IF FFIO=1       BR IF DATA CHAIN
12AA  F9B1    FILX 431                  434  SLION      BR IF FFI7=1       BR IF SLI IS ON
12AC  4886    FILX 432                                  I=I                TST FOR CCW COUNT 0
12AE  C4BA    FILX 433                  439  ZROSN2     BR IF ZNZ          BR IF ICL
12B0  DDB8    FILX 434    SLION         438  ZROSTA     BR IF FFI1=0       BR IF NOT CMND CHAIN
12B2  DA36    FILX 435                  437  SETA       BR IF HO BIT5=0    BR IF SEEK HAD MOTION
12B4  8640    FILX 436                  122  NTORST     BR                      GET NEXT CCW
12B6  85FE    FILX 437    SETA          212  GET8E      BR
12B8  2F07    FILX 438    ZROSTA                       H1=0               ZERO CHAN STATUD
12BA  4A52    FILX 439    ZROSN2                       RDH T DC,9A        T=0000
12BC  7A62    FILX 440                                  STH T DA,9C        ZERO SENSE BYTE 2
12BE  4DAF    FILX 441    CHNEND                        FFO=TO             RST FLAG REG.CC OFF PREVENTS DE&
              FILX 442  *                                                    MOD SEL UP LEVELS FROM TRAPPING
12C0  4FAF    FILX 443                                  FOP=TO             RST OP REG.PREVENT UNCHAINED INV
              FILX 444  *                                                    FILE OP FROM HD SWITCH IN CC TST
12C2  1E00    FILX 445                                  RST FIB K=80       CHAIN END RESET
12C4  1D00    FILX 446                                  RST FIA K=80       RESET TRAP LATCH
12C6  2E08    FILX 447                                  SET FIB K=40       RESET FLAG BITS 6,7
12C8  5C32    FILX 448                                  RDH P DA,8E        GET STATS
12CA  C3DE    FILX 449                  459  NOIPL      BR IF BA4=0        BR IF NOY IPL
12CC  CEC3    FILX 450                                  Z=HO*-KCC          MASK FOR CORRECT DEVICE STATUS
12CE  F0D3    FILX 451                  453  CHANTS     BR IF LZ=0         BR IF DEVICE STATUA OK
12D0  A7D8    FILX 452    STOP     DCLL 026  STOP       BR                 GO TO INTERLOCKED STOPWORD
12D2  0F85    FILX 453    CHANTS                        Z=H1*-K80          MASK FOR GOOD CHAN STATUS
12D4  C4D0    FILX 454                  452  STOP       BR IF ZNZ          BR IF CHAN STATUA BAD
12D6  2C07    FILX 455                                  PO=0               REMOVE ACTIVE STAT
12D8  7C32    FILX 456                                  STH P DA,8E
12DA  0212    FILX 457                                  RST MMSK K=11      RESET PRIORITY
12DC  A7F2    FILX 458             DCLL 040  PICKUP     BR                      BR TO LOAD PSW
12DE  2B43    FILX 459    NOIPL                         T1=0$K04           SET AUX ADDR 0004
12E0  7CA8    FILX 460                                  STH P AS,T+2       STORE I/O ADDR FOR CONT CONN
12E2  DC70    FILX 461                  468  STRINT     BR IF PO BIT1=0    BR IF NOT FROM SIO
12E4  7A32    FILX 462                                  STH T DA,8E        RESET STATS
12E6  C9EC    FILX 463                  466  NOPCI      BR IF FFI4=0       BR IF PCI OFF
12E8  3F85    FILX 464                                  H1=H1$K80          SET PCI IN ST&TUS
12EA  0D04    FILX 465                                  RST FIA K=20       RESET PCI LATCH
12EC  2410    FILX 466    NOPCI                         SET MODE K=01      SET CPU MODE FILE ZONE
12EE  A030    FILX 467             CCOM 135  STATUS     BR                      SET CCI
12F0  7EA0    FILX 468    STRINT                        STH H AS,T+0       STORE STATUS IN INTERRUPT BUFF
12F2  2D08    FILX 469                                  SET FIA K=40       SET INTERUPT LATCH
12F4  8602    FILX 470                  214  TRAPBR     BR                 BR TO CHECK IF IN TRAP
12F6  5EEF    FILX 471    EXAMIN                        HO=DS              GET DISK HARDWARE STATUS
12F8  1EC7    FILX 472                                  HO=HO*-KCC         REMOVE UNWHOLESOME BITS
12FA  C4D0    FILX 473                  452  STOP       BR IF ZNZ          IPL STOP ON MECHANICAL CHECK
```

```
ADDR     WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT          COMMENTS

12FC     5E42     FILX 474                                     RDH H DA,98        GET CCW ADDRESS
12FE     96DC     FILX 475               FILT 115   GOCAT      BR                 CONTINUE IPL
                  FILX 476   AEND
```

****
* CROSS REFERENCE FOR CSECT FILX *
****

```
FILX 007    FILT 060
FILX 015    FILX 022  FILX 023  FILX 053  FILX 054  FILX 060  FILX 078
FILX 028    FILX 030
FILX 029    FILX 025
FILX 041    FILX 027  FILX 029
FILX 043    FILX 031
FILX 045    FILX 045
FILX 052    FILX 009
FILX 056    FILX 056
FILX 060    FILX 017
FILX 061    FILX 063  FILX 064  FILX 065  FILX 069
FILX 062    FILX 013
FILX 063    FILX 067  FILX 068
FILX 064    FILX 062
FILX 067    FILX 021
FILX 069    FILX 052
FILX 078    FILX 010  FILX 014  FILX 018
FILX 080    FILX 079
FILX 089    FILX 044
FILX 092    FILX 093
FILX 093    FILX 091
FILX 095    FILX 279
FILX 097    FILT 073  FILX 092
FILX 100    FILX 098
FILX 101    FILX 099
FILX 106    FILX 103
FILX 109    FILX 104  FILX 106
FILX 113    FILX 109
FILX 115    FILX 046  FILX 111  FILX 113
FILX 116    FILX 265  FILX 271  FILX 378
FILX 121    FILX 118
FILX 122    FILT 164  FILX 436
FILX 123    FILX 372
FILX 132    FILX 127
FILX 133    FILX 129  FILX 130  FILX 131
FILX 140    FILX 133
FILX 141    FILX 121  FILX 376
FILX 142    FILX 117  FILX 120  FILX 275
FILX 149    FILX 012
FILX 153    FILX 149
FILX 156    FILX 150
FILX 158    FILX 161
FILX 159    FILX 066  FILX 188
FILX 161    FILX 156
FILX 162    FILT 158  FILX 028  FILX 155  FILX 157  FILX 173  FILX 174  FILX 185  FILX 187  FILX 192
```

****
* CROSS REFERENCE FOR CSECT FILX *
****

FILX 164	FILX 166								
FILX 165	FILX 016								
FILX 171	FILX 015	FILX 019	FILX 151	FILX 152	FILX 165	FILX 184	FILX 197	FILX 199	FILX 228
FILX 173	FILX 181								
FILX 174	FILX 154								
FILX 176	FILX 158	FILX 189	FILX 196						
FILX 177	FILX 163	FILX 172							
FILX 178	FILX 139	FILX 363							
FILX 181	FILX 020								
FILX 185	FILX 182	FILX 198							
FILX 188	FILX 191								
FILX 190	FILX 186								
FILX 193	FILX 164								
FILX 197	FILX 008								
FILX 203	FILX 200								
FILX 204	FILX 168								
FILX 205	FILX 061	FILX 072	FILX 175	FILX 183					
FILX 206	FILX 040	FILX 160	FILX 170	FILX 195	FILX 204				
FILX 208	FILX 206								
FILX 212	FILX 437								
FILX 213	FILT 077								
FILX 214	FILX 470								
FILX 216	FILX 214								
FILX 225	FILX 011								
FILX 229	FILX 227								
FILX 231	FILX 359								
FILX 238	FILX 255								
FILX 241	FILX 237								
FILX 244	FILX 241								
FILX 250	FILX 248								
FILX 254	FILX 263								
FILX 255	FILX 261								
FILX 259	FILX 240	FILX 257							
FILX 261	FILT 067								
FILX 266	FILX 264								
FILX 267	FILX 260								
FILX 271	FILX 274								
FILX 272	FILX 268	FILX 269							
FILX 276	FILX 267								
FILX 280	FILX 273								
FILX 282	FILX 303								
FILX 283	FILX 281	FILX 306							
FILX 288	FILX 266	FILX 270							
FILX 299	FILX 293	FILX 311	FILX 314						
FILX 300	FILX 100								
FILX 303	FILX 305								
FILX 304	FILX 302								
FILX 306	FILX 301								
FILX 307	FILX 298								

```
                                  **********************************
                                  * CROSS REFERENCE FOR CSECT FILX *
                                  **********************************

FILX 312     FILX 309
FILX 321     FILT 135
FILX 324     FILX 083
FILX 329     FILX 333  FILX 351
FILX 330     FILX 322
FILX 332     FILX 330
FILX 333     FILX 331
FILX 336     FILX 324
FILX 339     FILX 364
FILX 341     FILX 059
FILX 346     FILX 347
FILX 356     FILX 354
FILX 357     FILX 336
FILX 359     FILX 340  FILX 360
FILX 360     FILX 337
FILX 361     FILT 157  FILX 339  FILX 358
FILX 364     FILX 357
FILX 365     FILX 338
FILX 368     FILX 373
FILX 371     FILX 375
FILX 373     FILX 367
FILX 377     FILX 368  FILX 369  FILX 370
FILX 378     FILX 374
FILX 385     FILX 315
FILX 388     FILT 156
FILX 389     FILX 387
FILX 398     FILX 394
FILX 416     FILT 144
FILX 421     FILX 411
FILX 428     FILX 332
FILX 429     FILX 415
FILX 434     FILX 431
FILX 437     FILX 435
FILX 438     FILX 180  FILX 335  FILX 434
FILX 439     FILT 084  FILX 143  FILX 287  FILX 430  FILX 433
FILX 441     FILT 215
FILX 452     FILX 454  FILX 473
FILX 453     FILX 451
FILX 459     FILX 449
FILX 466     FILX 463
FILX 468     FILX 461
FILX 471     FILT 114
```

FINT DESCRIPTIVE TEXT

ENTRY POINTS

INTRPT — THIS IS THE NORMAL ENTRY INTO THE ROUTINE WHEN A FILE INTERRUPT IS DETECTED IN THE BSWI ROUTINE. THIS ROUTINE TESTS FOR FILE CONDITIONS BEFORE SETTING THE CSW INFORMATION INTO THE APPROPRIATE REGISTERS FOR THE CCOM ROUTINE.

SETCSW — THIS IS A BAL ENTRY TO SET THE CSW INFORMATION INTO THE CPU REGISTERS FOR STORING BY THE CCOM ROUTINE.

```
ADDR  WORD  SEQUENCE NO.  LABEL     NEXTSEQ    NEXTLABEL  STATEMENT            COMMENTS

            FINT 001   T                                  2311 FEATURE   INTERUPT ROUTINE FOR FILES
26FC  7CC2  FINT 002    INTRPT                              STH P DA,B8          SAVE CPU P REG
26FE  5F59  FINT 003                                        G1=H1                SAVE CPU H1 REG
2700  D98E  FINT 004                        011  TSTPCI     BR IF FFI5=0         BR IF NOT INETRUPT LATCH
2702  5A32  FINT 005                                       RDH T DA,8E          GET STATS AND IOA
2704  2A13  FINT 006                                       T0=0$K01             RESET STATS
2706  7A32  FINT 007                                       STH T DA,8E          STORE STATS
2708  9130  FINT 008                        050  SETCSW     BAL                  SET REGS FOR CPU CSW STORE
270A  A024  FINT 009    STUFCS    CCOM 127  CSWADR     BAL                  STORE CSW
270C  A748  FINT 010                        040  GETIOA     BR
270E  C996  FINT 011    TSTPCI              015  SETIOA     BR IF FFI4=0         BR IF NOT PCI
2710  2E07  FINT 012                                       H0=0                 SET DEVICE STATUS
2712  9138  FINT 013                        054  PCICSW     BAL                  SET CHAN STATUS PCI INFO
2714  A70A  FINT 014                        009  STUFCS     BR
2716  2B43  FINT 015    SETIOA                             T1=0$K04             SET AUX ADDR 0004
2718  4FBF  FINT 016                                       FOP=T1               SET CONTIN CONECT CLEAR IND
271A  5AA0  FINT 017                                       RDH T AS,T+0         GET BUFFERED IO ADDR FROM 0004
271C  2A13  FINT 018                                       T0=0$K01             RESET STATS
271E  7A32  FINT 019                                       STH T DA,8E          STORE STATS&I/O ADDR
2720  E1A8  FINT 020                        024  TSTDE      BR IF DASI6=0        BR IF NOT CUE
2722  0D02  FINT 021                                       RST FIA K=10         RESET CUB LATCH
2724  2E25  FINT 022                                       H0=0$K20             SET CUE FOR STATUS
2726  A740  FINT 023                        036  STOSTA     BR
2728  E5AF  FINT 024    TSTDE               027  RSTDE      BR IF DASI2=1        BR IF GATTED ATTENTION
272A  A5E4  FINT 025                 FILE 272  SNSSTA     BAL                  ERROR-NO REASON FOR INT-SET SNS
272C  A740  FINT 026                        036  STOSTA     BR
272E  8F90  FINT 027    RSTDE       FILE 288  SETDA      BAL                  SET DEV ADDR&SELECT MODULE
2730  4DBD  FINT 028                                       T1=P1L+T1H           BUILD DEVICE ADDRESS AND
2732  7A32  FINT 029                                       STH T DA,8E                 STORE IT IN AUX
2734  FABB  FINT 030                        033  SEKINC     BR IF DS7=1          BR IF SEEK INCOMPLETE
2736  A044  FINT 031                 FILE 344  RSTATT     BAL                  RESET GATED ATTN
2738  A73E  FINT 032                        035  SETDE      BR
273A  A044  FINT 033    SEKINC      FILE 344  RSTATT     BAL                  RESET GATED ATTN
273C  A5EC  FINT 034                FILE 276  SNSSTC     BAL                  SET SEEK CHECK SNS
273E  3E43  FINT 035    SETDE                              H0=H0$K04            SET DE FOR STATUS
2740  2F07  FINT 036    STOSTA                             H1=0                 ZERO CHAN STATUS
2742  2400  FINT 037    MODE                               SET MODE K=00        SET CPU ZONE&MODE
2744  2A07  FINT 038                                       T0=0
2746  A01E  FINT 039                 CCOM 123  STATOS     BAL                  ZERO CSW THEN STORE STATUS
```

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ   NEXTLABEL  STATEMENT          COMMENTS

2748    5A32    FINT 040    GETIOA                         RDH T DA,8E        GET I/O ADDRESS
274A    2A13    FINT 041                                   T0=0$K01           SET CHAN 1
274C    1000    FINT 042                                   RST S0
274E    5CC2    FINT 043                                   RDH P DA,B8        RESTORE P REG
2750    A43C    FINT 044                DCHN 104  FRMFIL   BR
                FINT 045  *             ********************************************************************
                FINT 046  *             *                                                                  **
                FINT 047  *             *                  SET CSW                    - SUBROUTINE         **
                FINT 048  *             *                                                                  **
                FINT 049  *             ********************************************************************
1130    2A07    FINT 050    SETCSW                         T0=0
1132    2B63    FINT 051                                   T1=0$K06           SET AUX ADDR 0006
1134    5EA0    FINT 052                                   RDH H AS,T+0       GET BUFFERED STATUS
1136    0D08    FINT 053                                   RST FIA K=40       RESET INTURRUPT LATCH
1138    526F    FINT 054    PCICSW                         D0=STP0            GET PROTECT KEY
113A    566B    FINT 055                                   D0=D0H
113C    2707    FINT 056                                   D1=0
113E    C9C2    FINT 057                  059  NOPCI       BR IF FFI4=0       BR IF NOT PCI INT
1140    3F85    FINT 058                                   H1=H1$K80          SET PCI IN STATUS
1142    2490    FINT 059    NOPCI                          SET MODE K=09      SET FILE MODE,FILE ZONE
1144    78A2    FINT 060                                   STH I DA,AC        BUFFER RESIDUAL COUNT
1146    0D04    FINT 061                                   RST FIA K=20       RESET PCI LATCH
1148    2400    FINT 062                                   SET MODE K=00      SET CPU MODE AND ZONE
114A    50A2    FINT 063                                   RDH U DA,AC        GET BUFFERED RESID COUNT
114C    5242    FINT 064                                   RDH V DA,98        GET CCW NEXT ADDR
114E    128E    FINT 065                                   RTN
```

****
* CROSS REFERENCE FOR CSECT FINT *
****

```
FINT 002    DCLT 053
FINT 009    FINT 014
FINT 011    FINT 004
FINT 015    FINT 011
FINT 024    FINT 020
FINT 027    FINT 024
FINT 033    FINT 030
FINT 035    FINT 032
FINT 036    FINT 023  FINT 026
FINT 040    FINT 010
FINT 050    FILE 017  FINT 008
FINT 054    FINT 013
FINT 059    FINT 057
```

OVLY DESCRIPTIVE TEXT

THE OVERLAY ROUTINE - OVLY- IS TO BE USED BY THE CUSTOMER WHEN A CHECK SUM OR STORAGE DATA ERROR OCCURES AND CAN NOT BE CLEARED BY PERFORMING A CSL. OVLY IS ENTERED BY CONTROL ADDRESS SETTING TO THE CTRL STORAGE ADDRESS 02BE WHICH IS THE LOCATION OF A UN-CONDITIONAL BRANCH TO THE FIRST INSTRUCTION OF THE OVLY ROUTINE.

THE OVLY ROUTINE WILL TRY TO RECONSTRUCT ALL BAD LOCATIONS OF MODULE ZERO BY DOING A COMPARISON AGAINST TWO OTHER DUPLICATES OF MODULE ZERO STORED IN VARIOUS AUX OR CTRL STORAGE LOCATIONS. THE LOCATION OF THE TWO COMPARISON MODULES IS DEPENDENT ON CORE SIZE AND FEATURES PRESENT. THE TWO COMPARISON MODULES ARE STORED AWAY EVERY TIME BCHK IS SUCCESFULLY EXECUTED.

IF A CSL CAN NOT BE PERFORMED AFTER THE EXECUTION OF THE OVLY ROUTINE THE CUSTOMER WILL HAVE TO CALL FOR SERVICE.

THE OVLY ROUTINE WILL NOT WORK IN ALL CASES (EX. OVLY ITSELF CREAMED). WHEN OVLY FAILS NORMAL TROUBLESHOOTING METHODS SHOULD BE USED.

****** CUSTOMER PROCEDURE TO RUN OVLY **********

1. IF A CUSTOMER CSL'S AND STILL HAS CHECK SUM OR STOR DATA SYSTEM CHECKS
2. SET MODE SWITCH TO ' W ' POSITION
3. DEPRESS SYSTEM RESET
4. SET CONSOLE SWITCHES ABCD TO X2BE
   (X=C FOR 48K MACHINES, X=8 FOR 32K MACHINES
   X=8 FOR 24K MACHINES, X=4 FOR 16K MACHINES)
5. DEPRESS CTRL ADDR SET
6. MODE SWITCH BACK TO PROCESS
7. DEPRESS START
8. DO ANOTHER CSL

ADDR	WORD	SEQUENCE NO.	LABEL	NEXTSEQ	NEXTLABEL	STATEMENT	COMMENTS
		OVLY 001	T			CUSTOMER ROUTINE TO CORRECT BAD MODULE ZERO WORDS	
30EC	3212	OVLY 002	IXSTRT			SET MMSK K=91	
30EE	3800	OVLY 003				SET DR K=80	SET ERROR STOP BYPASS
30F0	4652	OVLY 004				RDH D DC,9A	SET D TO ZERO
30F2	24C5	OVLY 006				G0=0$KC0	SET G TO C000
30F4	5759	OVLY 022				G1=D1	
30F6	2CD5	OVLY 024				P0=0$KD0	SET P TO D000
30F8	57D9	OVLY 042	OK			P1=D1	
30FA	4068	OVLY 043	OKRD			RDH U CS,D+2	READ CONTROL STORE INTO U
30FC	5248	OVLY 045				RDH V AS,G+2	READ AUX STORE INTO V
30FE	6021	OVLY 055				U0=U0<V0	COMPARE U0 TO V0
3100	C48A	OVLY 056		062	BAD	BR IF ZNZ	BR IF NO COMPARE
3102	6131	OVLY 057				U1=U1<V1	COMPARE U1 TO V1
3104	C48A	OVLY 058		062	BAD	BR IF ZNZ	BR IF NO COMPARE
3106	F229	OVLY 059	OKGO	087	GOODST	BR IF D07=1	LOOK FOR END OF 0 MODULE
3108	B0F8	OVLY 060		042	OK	BR	NOT END BRANCH TO CONTINUE
310A	50C8	OVLY 062	BAD			RDH U AS,P+2	CTRL STORE OR AUX STORE BAD
310C	6666	OVLY 072				D=D-2	ADJUST CTRL ADDRESS
310E	6201	OVLY 073				V0=V0<U0	COMPARE
3110	C49A	OVLY 074		079	BAD1	BR IF ZNZ	BR IF NO COMPARE
3112	6311	OVLY 075				V1=V1<U1	COMPARE
3114	C49A	OVLY 076		079	BAD1	BR IF ZNZ	BRANCH IF NO COMPARE
3116	6068	OVLY 077				STH U CS,D+2	DATA GOOD STORE INTO CTRL
3118	B106	OVLY 078		059	OKGO	BR	BRANCH TO CONTINUE
311A	4268	OVLY 079	BAD1			RDH V CS,D+2	COMPARE MOD BAD READ CTRL AGAIN
311C	6021	OVLY 080				U0=U0<V0	COMPARE
311E	C4A4	OVLY 081		084	ALLBAD	BR IF ZNZ	BRANCH IF NO COMPARE
3120	6131	OVLY 082				U1=U1<V1	COMPARE

```
ADDR    WORD  SEQUENCE NO.  LABEL     NEXTSEQ  NEXTLABEL  STATEMENT            COMMENTS

3122    C487    OVLY 083                 059    OKGO       BR IF Z=0            BRANCH IF NO COMPARE
3124    2820    OVLY 084     ALLBAD                        SET DR K=02          TURN ON CK SUM LIGHT AND HARD
                OVLY 085    *                                                                STOP LATCH
3126    2800    OVLY 086                                   SET DR K=00           RESET ERROR STOP BYPASS
3128    1212    OVLY 087     GOODST                        RST MMSK K=91        ALL OK STOP WITH NO CHK SUM ERR
312A    5007    OVLY 088     GDSTP                         STOP                 STOP GOOD
312C    B12A    OVLY 089                 088    GDSTP      BR                   LOCK IN
                                                          **********************************
                                                          * CROSS REFERENCE FOR CSECT OVLY *
                                                          **********************************

OVLY 002     BCHK 270
OVLY 042     OVLY 060
OVLY 059     OVLY 078  OVLY 083
OVLY 062     OVLY 056  OVLY 058
OVLY 079     OVLY 074  OVLY 076
OVLY 084     OVLY 081
OVLY 087     OVLY 059
OVLY 088     OVLY 089
```

	0	2	4	6	8	A	C	E	00--
0000	BCPL-003-8000	BCPL-004-0BAD	BCPL-004-0BAD	BCPL-004-0BAD	BCPL-004-0BAD	BCPL-004-0BAD	BCPL-004-0BAD	BCPL-005-80AC	0000
0010	BCPL-009-3210	BCPL-010-2610	BCPL-011-50CF	BCPL-012-51AF	BCPL-013-2413	BCPL-014-0CCF	BCPL-015-C4AD	BCPL-016-0CDF	0010
0020	BCPL-017-C48F	BCPL-018-0CEF	BCPL-019-C4E3	BCPL-021-0CFF	BCPL-022-C4A8	BCPL-023-80DE	BCPL-035-2486	BCPL-036-2507	0020
0030	BCPL-037-2B08	BCPL-038-C9B3	BCPL-039-4FAF	BCPL-040-2B04	BCPL-041-2B44	BCPL-042-C98A	BCPL-043-2B40	BCPL-044-CDBE	0030
0040	BCPL-045-2B23	BCPL-046-4FBF	BCPL-047-2B42	BCPL-048-FDC6	BCPL-049-5FBF	BCPL-050-C4CA	BCPL-051-2B48	BCPL-052-FDCF	0040
0050	BCPL-053-EDCE	BCPL-054-5FFF	BCPL-055-F05B	BCPL-056-7F48	BCPL-057-F05C	BCPL-058-6F48	BCPL-059-055D	BCPL-060-C4CC	0050
0060	BCPL-061-8100	BCPL-062-2406	BCPL-063-2617	BCPL-064-2783	BCPL-065-3775	BCPL-066-2507	BCPL-067-4066	BCPL-068-5EEF	0060
0070	BCPL-069-DE6E	BCPL-070-5B3F	BCPL-071-6348	BCPL-072-736A	BCPL-073-F0EE	BCPL-074-161B	BCPL-075-F0E7	BCPL-076-8100	0070
0080	BCPL-077-6EE3	BCPL-078-6EE3	BCPL-079-6EE3	BCPL-080-0E08	BCPL-081-0E04	BCPL-082-49EF	BCPL-083-4EDF	BCPL-084-4BFF	0080
0090	BCPL-085-2E63	BCPL-086-3E15	BCPL-087-4FEF	BCPL-088-2E43	BCPL-089-E598	BCPL-090-4EEF	BCPL-091-4BFF	BCPL-092-3D00	0090
00A0	BCPL-093-E5A1	BCPL-094-1D00	BCPL-095-50EF	BCPL-096-CA25	BCPL-097-C1A9	BCPL-098-810A	BCPL-099-240E	BCPL-100-2F04	00A0
00B0	BCPL-101-2507	BCPL-102-5FDF	BCPL-103-DAB8	BCPL-104-ED33	BCPL-105-2F10	BCPL-106-DAC6	BCPL-107-CAB9	BCPL-108-58FF	00B0
00C0	BCPL-109-5AEF	BCPL-110-4EF3	BCPL-111-6F48	BCPL-112-2B14	BCPL-113-055D	BCPL-114-C4B4	BCPL-115-8100	BCPL-117-FFFF	00C0
00D0	BCPL-117-FFFF	BCPL-117-FFFF	BCPL-117-FFFF	BCPL-117-FFFF	BCPL-117-FFFF	BCPL-117-FFFF	BCPL-118-FFFF	BCPL-128-2B07	00D0
00E0	BCPL-129-6A02	BCPL-130-2490	BCPL-131-2813	BCPL-132-2907	BCPL-133-4A86	BCPL-134-4C02	BCPL-135-2F15	BCPL-136-5FE9	00E0
00F0	BCPL-137-2D23	BCPL-138-4D9F	BCPL-139-8811	BCPL-140-54E8	BCPL-141-6488	BCPL-142-FA76	BCPL-143-0608	BCPL-144-88FE	00F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	00-- 01--
0100	CICY-064-A6D8	CICY-065-B055	CICY-073-814C	CICY-074-B075	CICY-062-834A	CICY-063-B009	CCOM-155-2400	CCOM-156-AA74	0100
0110	ETRP-020-2208	ETRP-021-A488	CLOG-148-B3F6	CLOG-149-6F11	CLOG-150-6E01	CLOG-151-6731	CLOG-152-6621	CLOG-153-A844	0110
0120	FILT-067-9852	FILT-068-8132	FILT-069-8132	FILT-070-FFAE	FILT-071-DBA0	FILT-072-5C52	FILT-073-860E	FILT-074-DBA0	0120
0130	FILT-075-CBA0	FILT-076-5E42	FILT-077-8600	CMPU-003-5B3A	CMPU-004-5F10	CMPU-005-4BF3	CMPU-006-5B75	CMPU-007-AE72	0130
0140	FILT-004-2222	FILT-018-8D87	CICY-129-2F07	CICY-130-C4DD	CICY-131-2707	CICY-132-128E	CICY-111-5098	CICY-112-50BB	0140
0150	CICY-113-C4C5	CICY-114-5EA8	CICY-115-56A0	CICY-116-617B	CICY-117-606F	CICY-118-6FAD	CICY-119-5298	CICY-120-52BB	0150
0160	CICY-121-C4C9	CICY-122-6AA4	CICY-123-56AA	CICY-124-637B	CICY-125-626F	CICY-126-56A0	CICY-127-67AD	CICY-128-128E	0160
0170	DCLD-010-2240	DCLD-011-8153	CCOM-064-50A2	CCOM-065-54B2	CCOM-066-5CC2	CCOM-067-5892	CCOM-068-2020	CCOM-069-128E	0170
0180	DCLE-010-2244	DCLE-011-C886	DCLE-012-0244	DCLE-013-3613	DCLE-014-CCA8	DCLE-015-D127	DCLE-016-5212	DCLE-017-5038	0180
0190	DCLE-018-008B	DCLE-019-F0A4	DCLE-020-5119	DCLE-021-C4A0	DCLE-022-5230	DCLE-023-0383	DCLE-024-F0A0	DCLE-025-95F4	0190
01A0	DCLE-026-5212	DCLE-027-95C2	DCLE-028-95FA	DCLE-029-A154	DCLE-030-8BE6	CCOM-174-2BB1	CCOM-175-7FA0	CCOM-176-128E	01A0
01B0	ETTR-018-2206	ETTR-019-5ACF	ETTR-020-B204	CMPU-032-5B3A	CMPU-033-4B73	CMPU-034-771A	CMPU-035-5B75	CMPU-036-AA1A	01B0
01C0	DCLD-013-81EC	DCLD-014-81DC	DCLD-015-F554	DCLD-016-F551	DCLD-017-7F3E	DCLD-018-5006	DCLD-019-C4E3	DCLD-020-0240	01C0
01D0	DCLD-021-2B48	DCLD-022-81CA	DCLD-023-7F3C	DCLD-024-5006	DCLD-025-C4E3	DCLD-026-0240	DCLD-027-5F3C	DCLD-028-5006	01D0
01E0	DCLD-029-C4DA	DCLD-030-C569	DCLD-031-1573	DCLD-032-0240	DCLD-033-3D00	DCLD-034-0240	DCLD-035-2307	DCLD-036-4F3F	01E0
01F0	DCLD-037-2B42	DCLD-038-D169	DCLD-039-3543	DCLD-040-0240	DCLA-402-52FF	DCLA-403-526C	DCLA-404-52EF	DCLA-405-51F2	01F0

	0	2	4	6	8	A	C	E	02--
0200	DCLA-410-C606	DCLA-412-41E5	DCLA-416-C609	DCLA-422-41EB	DCLA-423-42EF	DCLA-424-536C	DCLA-425-42FF	DCLA-426-128E	0200
0210	BWRP-008-887C	CMLT-172-3002	CMLT-173-3D25	CMLT-174-A752	CMLT-175-1D25	CMLT-176-3002	CMLT-177-A754	CMLT-178-89C4	0210
0220	BMCK-003-3262	BMCK-004-9028	FILE-258-5C52	FILE-259-F939	FILE-260-EC38	FILE-261-2C07	FILE-262-2D43	FILE-263-5EC0	0220
0230	FILE-264-4FCF	FILE-265-6F11	FILE-266-C4B9	FILE-267-4FDF	FILE-268-5CC2	FILE-269-128E	BDIA-075-2810	BDIA-080-F56C	0230
0240	BDIA-018-2810	BDIA-030-3210	BDIA-048-2400	BDIA-053-F8CB	BDIA-054-B7C9	BDIA-056-2810	BDIA-057-25F3	BDIA-061-C54F	0240
0250	BDIA-062-2810	BDIA-066-D553	BDIA-067-2810	BDIA-072-E53C	BCHK-257-0C2B	BCHK-258-C4E4	BCHK-259-0D8D	BCHK-260-C4E4	0250
0260	BCHK-261-2DB5	BCHK-262-2DEB	BCHK-263-42C0	BCHK-264-6A21	BCHK-265-6B31	BCHK-266-128E	BDIA-083-2810	BDIA-084-9FFC	0260
0270	CCOM-036-55B3	CCOM-037-52A8	CCOM-038-50AA	CCOM-039-55BB	CCOM-040-56A8	CCOM-041-5EAA	CCOM-042-100E	CCOM-043-128E	0270
0280									0280
0290									0290
02A0									02A0
02B0								BCHK-270-B0EC	02B0
02C0	CICY-185-ACD4	CICY-186-AD76	CICY-187-AEA6	CICY-188-A874	CICY-189-AD88	CICY-190-A91C	CICY-191-ADB2	CICY-192-A942	02C0
02D0	CICY-193-ACC0	CICY-194-A3EE	CICY-195-88E8	CICY-196-969C	CICY-197-8978	CICY-198-83B4	CICY-199-AC4E	CICY-200-AC4E	02D0
02E0	CICY-251-83B4	CICY-252-AE56	CICY-253-AE7C	CICY-254-AE20	CICY-255-A944	CICY-256-A828	CICY-257-A988	CICY-258-AD62	02E0
02F0	CICY-259-83B4	CICY-260-83B4	CICY-261-83B4	CICY-262-83B4	CICY-263-ADD0	CICY-264-ADD0	CICY-265-AA78	CICY-266-AA78	02F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	02-- 03--
0300	FILT-025-96A2	FILT-026-96A2	FILT-027-830A	FILT-028-2F7B	FILT-029-5EE4	FILT-030-1D00	FILT-031-5CF8	FILT-032-A967	0300
0310	FILT-033-0C8B	FILT-034-F0A2	FILT-035-5EF0	FILT-036-0F83	FILT-037-F0DE	FILT-038-5CF8	FILT-039-A967	FILT-040-0C8B	0310
0320	FILT-041-C4E1	FILT-042-CDA9	FILT-043-4FCF	FILT-044-F0E1	FILT-045-5AF8	FILT-046-5CF8	FILT-047-58F8	FILT-048-0C83	0320
0330	FILT-049-F0E4	FILT-050-4886	FILT-051-C4E5	FILT-052-7E42	FILT-053-FC3E	FILT-054-FBBF	FILT-055-2A07	FILT-057-CDED	0330
0340	FILT-058-4DCF	FILT-059-5C52	FILT-060-BB9D	CICY-090-2F07	CICY-091-128E	CICY-081-5098	CICY-082-50BB	CICY-083-C4C7	0340
0350	CICY-084-5EA8	CICY-085-56A0	CICY-086-617B	CICY-087-606F	CICY-088-6FAD	CICY-089-128E	DCLA-228-8A8C	FILT-078-5E42	0350
0360	FILT-079-3F73	FILT-080-5EE4	FILT-081-7E42	FILT-082-2F25	FILT-083-2E07	FILT-084-928A	FILT-061-4DCF	FILT-062-2C07	0360
0370	FILT-063-2D35	FILT-064-3D43	FILT-065-5EC0	FILT-066-8BA3	DYPE-191-5E29	DYPE-192-2791	DYPE-193-7C68	DYPE-194-7468	0370
0380	DYPE-195-7268	DYPE-196-7068	DYPE-197-2771	DYPE-198-128E	BCHK-324-07FF	BCHK-324-C48A	BCHK-324-8800	BCHK-324-4E08	0380
0390	CDVD-002-E195	CDVD-003-3485	CDVD-004-8AC8	CDVD-005-B2C2	BCHK-327-00C8	BCHK-327-0000	BCHK-330-4E08	BCHK-330-128E	0390
03A0	CMLT-181-3002	CMLT-182-A752	CMLT-183-3002	CMLT-184-B472	BCHK-334-0000	BCHK-334-0000	BCHK-334-0000	BCHK-334-0000	03A0
03B0	CICY-075-6884	CICY-076-6884	CICY-077-2B13	CICY-078-89FC	BCHK-349-2FF7	BCHK-349-2F1B	BCHK-349-C4BC	BCHK-349-0F01	03B0
03C0	BDIA-347-C4C0	BDIA-348-4AC2	BDIA-349-4CD2	BDIA-350-4EF2	BDIA-351-2A1B	BDIA-352-2C1B	BDIA-353-2E1B	BDIA-354-E0F2	03C0
03D0	BDIA-355-6AC2	BDIA-356-6CD2	BDIA-357-6EF2	BDIA-358-83B8	BDIA-331-4812	BDIA-332-3935	BDIA-333-292B	BDIA-334-68E2	03D0
03E0	BDIA-335-2A25	BDIA-336-2BF5	BDIA-337-3B73	BDIA-338-2C25	BDIA-339-2D15	BDIA-340-3DB3	BDIA-341-2E05	BDIA-342-2F13	03E0
03F0	BDIA-343-83D0	BDIA-373-2505	BDIA-374-251B	BDIA-375-F4FD	BDIA-376-C4F4	BDIA-377-5007	BDIA-381-2413	BDIA-382-883C	03F0

	0	2	4	6	8	A	C	E	04--
0400	CICY-225-B412	CICY-226-A85A	CICY-227-A9BA	CICY-228-AE44	CICY-229-A84C	CICY-230-A816	CICY-231-AB72	CICY-232-A958	0400
0410	CICY-233-B420	CICY-239-83B4	CICY-241-83B4	CICY-246-83B4	CICY-247-8886	CICY-248-8886	CICY-249-8886	CICY-250-8886	0410
0420	CSFT-087-0002	CSFT-088-D02C	CSFT-089-611B	CSFT-090-600D	CSFT-091-633D	CSFT-092-622D	CSFT-093-677D	CSFT-094-666D	0420
0430	CSFT-095-6FFD	CSFT-096-6EED	CSFT-097-E4B8	CSFT-098-3000	CSFT-099-6AAD	CSFT-100-2DFB	CSFT-101-F0A0	CSFT-102-128E	0430
0440	BDIA-122-5007	DCLC-116-2B48	DCLC-118-AF70	BSWI-007-2F07	BSWI-008-3600	BSWI-009-00C0	BSWI-010-81AA	BSWI-012-5AEF	0440
0450	BSWI-013-1EFF	BSWI-014-C4F7	BSWI-024-5E09	BSWI-025-6EE3	BSWI-026-6EE3	BSWI-027-6E03	BSWI-034-2B55	BSWI-035-50B8	0450
0460	BSWI-036-52BA	BSWI-037-2002	BSWI-038-72E9	BSWI-039-71A9	BSWI-040-70A9	BSWI-042-42AF	BSWI-044-70B8	BSWI-045-72BA	0460
0470	BSWI-047-9243	BSWI-049-F5F9	BSWI-050-2608	BSWI-051-AB04	BSWI-052-E976	BSWI-053-8A54	FILE-008-7CC2	FILE-009-2480	0470
0480	FILE-010-C62A	FILE-011-E013	FILE-012-F020	FILE-013-D9A0	FILE-014-6711	FILE-015-C4A0	FILE-016-7032	FILE-017-9130	0480
0490	FILE-018-A024	FILE-019-5DBF	FILE-020-1BC5	FILE-021-D9A3	FILE-022-3210	FILE-023-2D04	FILE-024-4DBF	FILE-025-1210	0490
04A0	FILE-026-AFCA	FILE-027-F5BD	FILE-028-2E07	FILE-029-D9C0	FILE-030-810C	FILE-031-3065	FILE-032-7032	FILE-033-E023	04A0
04B0	FILE-034-F5BD	FILE-035-E1C3	FILE-036-F056	FILE-037-DBFD	FILE-038-8224	FILE-039-DBFC	FILE-040-2E55	FILE-041-2D02	04B0
04C0	FILE-042-855E	FILE-043-DBCC	FILE-044-8224	FILE-045-DBBD	FILE-046-2E43	FILE-047-4FEF	FILE-048-2E25	FILE-049-F053	04C0
04D0	FILE-050-2E35	FILE-051-0D02	FILE-052-855E	FILE-053-7242	FILE-054-42EF	FILE-055-5A30	FILE-056-0A4B	FILE-057-C4E4	04D0
04E0	FILE-058-8FA6	FILE-059-910C	FILE-060-DBEB	FILE-061-8224	FILE-062-DBBD	FILE-063-0A3B	FILE-064-C4FD	FILE-065-0A1D	04E0
04F0	FILE-066-E0F6	FILE-067-0A7B	FILE-068-F0FD	FILE-070-0133	FILE-074-F0FD	FILE-075-850E	FILE-076-5ECF	FILE-077-1CA3	04F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	04-- 05--
0500	FILE-078-C489	FILE-079-2C43	FILE-080-A5E6	FILE-081-855A	FILE-082-8FA6	FILE-083-EE95	FILE-084-DE99	FILE-085-AEEC	0500
0510	FILE-086-2A45	FILE-087-8558	FILE-088-2C85	FILE-089-8504	FILE-090-2E15	FILE-091-FAAB	FILE-092-CEBE	FILE-093-E5B7	0510
0520	FILE-094-F025	FILE-095-9068	FILE-096-49AF	FILE-097-5CC2	FILE-098-810C	FILE-099-AEEC	FILE-100-0A3B	FILE-101-F0C6	0520
0530	FILE-102-0A1D	FILE-103-E0C6	FILE-104-910C	FILE-105-A044	FILE-106-2E43	FILE-107-F03F	FILE-108-3E15	FILE-109-A052	0530
0540	FILE-110-F061	FILE-111-AEEC	FILE-112-856A	FILE-113-2A13	FILE-114-E5D6	FILE-115-A044	FILE-116-A5EC	FILE-117-3E43	0540
0550	FILE-118-F05B	FILE-119-3E15	FILE-120-855A	FILE-121-A044	FILE-122-A5EC	FILE-123-2B43	FILE-124-70A0	FILE-125-F068	0550
0560	FILE-126-2400	FILE-127-5CC2	FILE-128-2F07	FILE-129-A01E	FILE-130-E073	FILE-131-234B	FILE-132-5630	FILE-133-2F85	0560
0570	FILE-134-C275	FILE-135-2F07	FILE-136-2A07	FILE-137-C3FF	FILE-138-2400	FILE-139-5CC2	FILE-140-A030	FILE-141-A7D8	0570
0580	FILX-149-DF88	FILX-150-DF8E	FILX-151-85AC	FILX-152-85AC	FILX-153-2D10	FILX-154-E833	FILX-155-859A	FILX-156-C819	0580
0590	FILX-157-FC1A	FILX-158-0D36	FILX-159-2C43	FILX-160-85F2	FILX-161-D193	FILX-162-2F15	FILX-163-85B8	FILX-164-DFD8	0590
05A0	FILX-165-85AC	FILX-166-859E	FILX-167-2C23	FILX-168-FFEE	FILX-169-2C83	FILX-170-CFF2	FILX-171-2F07	FILX-172-85B8	05A0
05B0	FILX-173-F81A	FILX-174-D19A	FILX-175-FD70	FILX-176-2F43	FILX-177-2EA5	FILX-178-7E52	FILX-179-2E23	FILX-180-92B8	05B0
05C0	FILX-181-DFB0	FILX-182-DFC8	FILX-183-EFF1	FILX-184-85AC	FILX-185-E89B	FILX-186-E853	FILX-187-D81A	FILX-188-ED14	05C0
05D0	FILX-189-85B6	FILX-190-2D10	FILX-191-D1CF	FILX-192-859A	FILX-193-2E08	FILX-194-2C15	FILX-195-DD73	FILX-196-85B6	05D0
05E0	FILX-197-85AC	FILX-198-DFC8	FILX-199-85AC	FILX-200-DC6C	FILX-201-3E00	FILX-202-0E04	FILX-203-2C33	FILX-204-CFF2	05E0
05F0	FILX-205-2C07	FILX-206-D6F7	FILX-207-221E	FILX-208-3D0C	FILX-209-7C52	FILX-210-2C15	FILX-211-4BCF	FILX-212-1D00	05F0

	0	2	4	6	8	A	C	E	06--
0600	FILX-213-5C32	FILX-214-D686	FILX-215-0222	FILX-216-2C85	FILX-217-7C32	FILX-218-021E	FILX-219-810C	FILX-097-0D83	0600
0610	FILX-098-FC94	FILX-099-ED16	FILX-100-98A6	FILX-101-3D13	FILX-102-2C25	FILX-103-C920	FILX-104-FD26	FILX-105-3DF3	0610
0620	FILX-106-FD26	FILX-107-3DD3	FILX-108-1D15	FILX-109-CD2F	FILX-110-3D85	FILX-111-DD32	FILX-112-1DFD	FILX-113-DD33	0620
0630	FILX-114-3DA5	FILX-115-7C52	FILX-116-2F45	FILX-117-CDDB	FILX-118-F9BF	FILX-119-4886	FILX-120-C4DA	FILX-121-DDD8	0630
0640	FILX-122-2E08	FILX-123-0E04	FILX-124-5E42	FILX-125-5CFC	FILX-126-0CC3	FILX-127-C4D5	FILX-128-0C3B	FILX-129-F0DE	0640
0650	FILX-130-DEDE	FILX-131-EEDF	FILX-132-830A	FILX-140-8D87	FILX-141-2F07	FILX-142-2EC3	FILX-143-92BA	FILX-133-EDD7	0650
0660	FILX-134-2F68	FILX-135-58F8	FILX-136-7E42	FILX-137-2F07	FILX-138-2E65	FILX-139-85BA	FILE-246-2308	FILE-247-2308	0660
0670	FILE-248-2308	FILE-249-2308	FILE-250-128E	FILE-251-2112	FILE-252-5ECF	FILE-253-6CD1	FILE-254-C4F5	FILE-255-90FC	0670
0680	FILX-007-9758	FILX-008-8FFB	FILX-009-EFB8	FILX-010-EFE2	FILX-011-9874	FILX-012-8F9B	FILX-013-EFCC	FILX-014-EFE2	0680
0690	FILX-015-85AC	FILX-016-8FBB	FILX-017-EFC8	FILX-018-EFE2	FILX-019-85AC	FILX-020-8FDB	FILX-021-EFD6	FILX-022-EF91	0690
06A0	FILX-023-DF91	FILX-024-5E32	FILX-025-FFF6	FILX-026-3E0C	FILX-027-CD2C	FILX-028-859A	FILX-041-1E45	FILX-042-7E32	06A0
06B0	FILX-043-2C07	FILX-044-FFE7	FILX-045-F8B4	FILX-046-8632	FILX-052-FFDB	FILX-053-DF91	FILX-054-CD11	FILX-055-3E00	06B0
06C0	FILX-056-F5C1	FILX-057-2C65	FILX-058-4DCF	FILX-059-872A	FILX-060-FF90	FILX-061-85F0	FILX-062-E851	FILX-063-F84A	06C0
06D0	FILX-064-FFCB	FILX-065-D1CA	FILX-066-8594	FILX-067-FFCE	FILX-068-DC4E	FILX-069-DC4A	FILX-070-3E00	FILX-071-0E04	06D0
06E0	FILX-072-85F0	FILX-078-DF91	FILX-079-9058	FILX-089-5DB8	FILX-090-5886	FILX-091-CDEF	FILX-092-860E	FILX-093-C4EC	06E0
06F0	FILX-094-7C52	FILX-095-5E42	FILX-096-830A	FILX-029-DE2D	FILX-030-F5AB	FILX-031-EC31	FILX-032-2C33	FILX-033-4FCF	06F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	06-- 07--
0700	FILX-034-5DDD	FILX-035-3DB5	FILX-036-5E42	FILX-037-3E00	FILX-038-0E04	FILX-039-2C45	FILX-040-85F2	FILX-321-1D00	0700
0710	FILX-322-E976	FILX-323-5C52	FILX-324-F5A0	FILX-325-2D23	FILX-326-3D15	FILX-327-1D00	FILX-328-4BDF	FILX-329-0222	0710
0720	FILX-336-FFF1	FILX-337-CBE9	FILX-338-DBCC	FILX-339-C96B	FILX-340-FFF4	FILX-341-2E08	FILX-342-2D07	FILX-343-4BDF	0720
0730	FILX-344-2D23	FILX-345-4EDF	FILX-346-2D1B	FILX-347-F0B4	FILX-348-4BDF	FILX-349-5E32	FILX-350-1D00	FILX-351-D69F	0730
0740	FILX-352-2E85	FILX-353-7E32	FILX-354-C3C9	FILX-355-810C	FILX-356-96D8	FILX-364-DBA6	FILX-365-5E32	FILX-366-1E45	0740
0750	FILX-367-FFE0	FILX-368-F9DC	FILX-369-CDDD	FILX-370-DDDC	FILX-371-7E32	FILX-372-8642	FILX-377-7E32	FILX-378-8634	0750
0760	FILX-373-F5D3	FILX-374-CDDF	FILX-375-DDD9	FILX-376-8658	FILX-360-D974	FILX-361-2E25	FILX-362-2F43	FILX-363-85BA	0760
0770	FILX-357-CBCA	FILX-358-E96B	FILX-359-9860	FILX-330-CDFB	FILX-331-F9FD	FILX-332-92A4	FILX-333-DD9F	FILX-334-2E83	0770
0780	FILX-335-92B8	DYPE-003-FFA3	DYPE-004-FAA5	DYPE-005-9816	DYPE-006-A566	DYPE-007-C826	DYPE-008-CE92	DYPE-009-3E85	0780
0790	DYPE-010-0F10	DYPE-011-1F28	DYPE-012-3C83	DYPE-013-8FBE	DYPE-014-CE9C	DYPE-015-2F04	DYPE-016-8378	DYPE-018-FEDC	0790
07A0	DYPE-019-9A0C	DYPE-020-A6B8	DYPE-021-947A	DYPE-022-D13D	DYPE-023-CE2D	DYPE-024-2E43	DYPE-025-C413	DYPE-027-2485	07A0
07B0	DYPE-028-DE93	DYPE-029-2E07	DYPE-030-CE8F	DYPE-031-FEA1	DYPE-032-3423	DYPE-033-8792	DYPE-034-C471	DYPE-036-2E07	07B0
07C0	DYPE-037-EEC6	DYPE-038-3D45	DYPE-039-B36E	DYPE-040-8FBA	DYPE-041-DFF8	DYPE-043-C05F	DYPE-044-5F38	DYPE-045-4FFF	07C0
07D0	DYPE-046-0F20	DYPE-047-5006	DYPE-048-C4DA	DYPE-049-C57B	DYPE-050-3483	DYPE-051-837A	DYPE-052-AB20	DYPE-053-1483	07D0
07E0	DYPE-054-E567	DYPE-055-3445	DYPE-056-15C5	DYPE-057-E145	DYPE-058-5DD9	DYPE-059-C4F1	DYPE-060-15C5	DYPE-061-3E23	07E0
07F0	DYPE-062-D57B	DYPE-063-3EC3	DYPE-064-3485	DYPE-065-8792	DYPE-066-A1D2	DYPE-067-1485	DYPE-068-8378	DYPE-070-A598	07F0

	0	2	4	6	8	A	C	E	08--
0800	BDIA-280-4E02	BDIA-281-2E1B	BDIA-282-2F1D	BDIA-283-CA0D	BDIA-284-6E02	BDIA-285-8388	BDIA-302-E1A3	BDIA-303-2020	0800
0810	BDIA-304-E190	BDIA-305-2EC5	BDIA-306-3E43	BDIA-307-2F85	BDIA-308-3FA3	BDIA-309-6E12	BDIA-310-2EF7	BDIA-311-3F89	0810
0820	BDIA-312-8802	BDIA-313-0020	BDIA-314-E1A5	BDIA-315-D1AC	BDIA-316-0040	BDIA-317-83D8	BDIA-318-2040	BDIA-319-D1AE	0820
0830	BDIA-320-4E22	BDIA-321-6E12	BDIA-322-3EC9	BDIA-323-5EF1	BDIA-324-8802	BDIA-385-6443	BDIA-386-2505	BDIA-387-7543	0830
0840	BDIA-388-3808	BDIA-389-050D	BDIA-390-3800	BDIA-391-DAC6	BDIA-392-1212	BDIA-393-2800	BDIA-394-C43A	BDIA-399-2440	0840
0850	BDIA-400-886E	BDIA-401-0E3F	BDIA-402-FED4	BDIA-403-4E80	BDIA-404-6E80	BDIA-405-FADA	BDIA-406-0E3F	BDIA-407-EADE	0850
0860	BDIA-408-1212	BDIA-409-886E	BDIA-410-0F1F	BDIA-411-FEE6	BDIA-412-4E80	BDIA-413-6E80	BDIA-414-02E2	BDIA-415-4E80	0860
0870	BDIA-416-3804	BDIA-417-6E80	BDIA-418-CAF4	BDIA-419-3800	BDIA-420-128E	BWRP-050-97F5	BWRP-011-D4FA	BWRP-012-979D	0870
0880	DCLA-121-AFF2	DCLA-122-9C28	DCLA-123-A25E	DCLA-002-10CE	DCLA-003-F90C	DCLA-004-ADFC	DCLA-005-2C07	DCLA-006-2B45	0880
0890	DCLA-007-3B83	DCLA-008-46A6	DCLA-009-17FD	DCLA-010-804F	CSTS-112-83B4	ALDP-258-2400	ALDP-259-AAB1	ALDP-260-0610	0890
08A0	ALDP-261-AB20	ALDP-262-3BE9	ALDP-263-76AA	ALDP-264-72AA	ALDP-265-74AA	ALDP-266-70A0	ALDP-267-2B2D	ALDP-268-7EA0	08A0
08B0	ALDP-269-2440	ALDP-270-7872	ALDP-271-2400	ALDP-272-E33B	ALDP-273-1600	ALDP-274-AB20	ALDP-241-2315	ALDP-242-3353	08B0
08C0	ALDP-243-3F00	ALDP-244-C4CE	ALDP-246-2F20	ALDP-247-3F02	ALDP-248-1683	ALDP-249-C4D0	ALDP-251-2345	ALDP-252-2F0A	08C0
08D0	ALDP-253-EEA2	ALDP-254-1F2A	ALDP-255-E31A	ALDP-256-FA9B	ALDP-257-9302	CBIN-042-B3F6	CBIN-043-B460	CBIN-044-AC1C	08D0
08E0	BPSW-069-2B53	BPSW-070-2407	BPSW-071-6884	BPSW-072-89FE	CBIN-034-AE06	CBIN-035-AC1A	BDIA-269-D1ED	BDIA-270-E1EF	08E0
08F0	BDIA-271-2E87	BDIA-272-2F05	BDIA-273-6E12	BDIA-274-6E22	BDIA-275-3EE9	BDIA-276-2F77	BDIA-277-8802	BDIA-008-8240	08F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	08-- 09--
0900	CMLT-134-89C4	CMLT-135-8920	CMLT-136-A37A	CMLT-137-B46C	CMLT-138-B472	CMLT-139-B46E	CMLT-140-A200	CMLT-141-A372	0900
0910	CMLT-142-A378	CMLT-143-903A	CMLT-144-8212	CMLT-145-83A0	CMLT-146-AFFC	CMLT-147-83A2	CMLT-148-821A	CMLT-149-3002	0910
0920	CMLT-150-A752	CMLT-151-89C4	CFLS-034-4606	CFLS-035-45BB	CFLS-036-B42E	CFLS-037-7E78	CFLS-038-7878	CFLS-039-F435	0920
0930	CFLS-040-7278	CFLS-041-7070	CFLS-042-817A	CFLS-043-AA76	DCLA-069-5EB8	DCLA-071-7EF2	DCLA-072-0EF5	DCLA-073-F0C3	0930
0940	DCLA-074-B042	DCLA-075-AB43	DCLA-081-52BA	DCLA-089-0249	DCLA-091-F4C1	DCLA-092-0383	DCLA-093-F0C0	DCLA-095-5009	0940
0950	DCLA-096-C4D4	DCLA-097-8A7A	DCLA-099-001B	DCLA-100-C4E8	DCLA-105-5632	DCLA-107-2BA1	DCLA-108-5FAA	DCLA-109-5FE1	0950
0960	DCLA-110-6F11	DCLA-111-E0E7	DCLA-120-8011	DCLA-125-847C	DCLA-133-95BC	CFCO-078-7EA8	CFCO-079-78A8	CFCO-080-F475	0960
0970	CFCO-081-72A8	CFCO-082-70A0	CFCO-083-1B73	CFCO-084-128E	CMLT-014-AE06	CMLT-015-9814	CMLT-016-2907	CMLT-017-C604	0970
0980	CMLT-018-29F7	CMLT-019-2C13	CMLT-020-0275	CMLT-021-E08B	CMLT-022-3C23	CMLT-023-5989	CMLT-024-4A86	CMLT-025-00C0	0980
0990	CMLT-026-7CF2	CMLT-027-4C52	CMLT-028-2D1D	CMLT-029-2D8B	CMLT-030-CD29	CMLT-031-78C8	CMLT-032-7AC8	CMLT-033-76C8	0990
09A0	CMLT-034-7EC8	CMLT-035-9AFA	CMLT-036-9AFA	CMLT-037-8994	CMLT-038-2D07	CMLT-039-48C6	CMLT-040-4AC6	CMLT-041-44C6	09A0
09B0	CMLT-042-46C6	CMLT-043-4E06	CMLT-044-40C6	CMLT-045-2C13	CMLT-046-C5BE	CMLT-047-2F1B	CMLT-048-F4D3	CMLT-049-2D15	09B0
09C0	CMLT-050-1002	CMLT-051-BB03	CMLT-052-DD53	CMLT-053-C5CC	CMLT-054-2F1D	CMLT-055-F4D3	CMLT-056-2D55	CMLT-057-1002	09C0
09D0	CMLT-058-BF03	CMLT-059-2C2B	CMLT-060-AF4A	CMLT-061-C838	CMLT-063-C1DC	CMLT-064-A354	CMLT-065-54F2	CMLT-069-C5E9	09D0
09E0	CMLT-070-E068	CMLT-071-3002	CMLT-072-2D15	CMLT-073-A752	CMLT-074-8176	CMLT-075-2A07	CMLT-076-55BB	CMLT-077-72A8	09E0
09F0	CMLT-078-7EAA	CMLT-079-F47A	CMLT-080-1B15	CMLT-081-76A8	CMLT-082-70AA	CMLT-083-AA76	BPSW-003-2F07	BPSW-004-2125	09F0

	0	2	4	6	8	A	C	E	0A--
0A00	BPSW-005-21EB	BPSW-006-2007	BPSW-007-420F	BPSW-008-5CEB	BPSW-009-3E45	BPSW-010-4DE5	BPSW-011-E193	BPSW-012-5892	0A00
0A10	BPSW-013-54B2	BPSW-014-C416	BPSW-015-2E4D	BPSW-016-D41A	BPSW-017-2E4D	BPSW-018-781A	BPSW-019-7E1A	BPSW-020-7A1A	0A10
0A20	BPSW-021-5E82	BPSW-022-7E10	BPSW-023-214D	BPSW-024-5C18	BPSW-025-44CF	BPSW-027-42DF	BPSW-029-7C82	BPSW-030-6004	0A20
0A30	BPSW-031-5E18	BPSW-032-5810	BPSW-033-2020	BPSW-034-4ED3	BPSW-035-5ECB	BPSW-036-1CC5	BPSW-037-FC40	BPSW-038-3C45	0A30
0A40	BPSW-039-2A07	BPSW-040-7CC2	BPSW-041-4FDF	BPSW-042-E955	BPSW-054-2020	BPSW-055-F1D8	BPSW-059-AB71	BPSW-061-09E3	0A40
0A50	BPSW-062-F0DA	BPSW-063-AA06	BPSW-066-0004	BPSW-067-F1D5	BPSW-068-8448	BPSW-064-2B63	BPSW-065-88E2	CSTS-113-83B4	0A50
0A60	DCLA-392-8CC8	DCLA-394-9F24	DCLA-395-9F24	DCLA-400-B31C	ERRQ-091-2A15	ERRQ-092-F774	ERRQ-093-2406	ERRQ-094-2BF5	0A60
0A70	ERRQ-095-5CAA	ERRQ-096-128E	ERRQ-097-2402	ERRQ-098-2BB1	ERRQ-099-C4F0	DCLA-141-2543	DCLA-142-4E66	DCLA-143-1FCB	0A70
0A80	DCLA-144-271D	DCLA-145-5BE8	DCLA-146-6B11	DCLA-147-C48D	DCLA-148-DB01	DCLA-173-95BC	DCLA-251-5E60	DCLA-253-E024	0A80
0A90	DCLA-274-8F7A	DCLA-275-07FD	DCLA-276-E0C0	DCLA-277-A46C	DCLA-282-C41E	DCLA-283-D123	DCLA-284-DA23	DCLA-285-4E52	0A90
0AA0	DCLA-286-A030	DCLA-287-810C	DCLA-305-F037	DCLA-306-DB35	DCLA-320-CE3F	DCLA-324-5669	DCLA-325-E0B1	DCLA-326-81F8	0AA0
0AB0	DCLA-327-3773	DCLA-329-95F4	DCLA-369-AECA	DCLA-354-DB3C	DCLA-365-CE34	DCLA-366-DF35	DCLA-367-3613	DCLA-368-8765	0AB0
0AC0	DCLA-278-D518	DCLA-279-15C5	DCLA-280-7460	DCLA-281-8A9E	CCOM-029-55B3	CCOM-030-56A8	CCOM-031-5EAA	CCOM-032-55BB	0AC0
0AD0	CCOM-033-0008	CCOM-034-128E	CSTS-068-8ADA	CSTS-069-7098	CSTS-070-AA76	CSTS-071-500F	CSTS-072-511F	CSTS-073-128E	0AD0
0AE0	BWRP-051-9D74	BWRP-053-96B0	BWRP-057-B1FC	BWRP-061-B1FC	BWRP-065-B1FC	BWRP-073-5007	BWRP-075-B1FC	BWRP-079-98D6	0AE0
0AF0	5C07	CFAD-007-3B35	CFAD-008-896A	CFAD-009-5E79	CFAD-010-5AE2	CFAD-011-842E	CFAD-012-2C07	CFAD-013-5E69	0AF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	0A-- 0B--
0B00	CFAD-014-5ED9	CFAD-015-2040	CFAD-016-7671	CFAD-018-E48E	CFAD-019-F491	CFAD-020-56C9	CFAD-021-8B16	CFAD-022-F48B	0B00
0B10	CFAD-023-0040	CFAD-025-57E9	CFAD-026-7C61	CFAD-027-F01A	CFAD-028-1C8D	CFAD-029-CC1E	CFAD-030-3002	CFAD-031-1C85	0B10
0B20	CFAD-032-E0A5	CFAD-033-2CF3	CFAD-034-F0BF	CFAD-035-D1B6	CFAD-036-3B35	CFAD-037-B42E	CFAD-038-A656	CFAD-039-896A	0B20
0B30	CFAD-040-5AE2	CFAD-041-B42E	CFAD-042-8B40	CFAD-043-1E85	CFAD-044-CD3C	CFAD-045-3E85	CFAD-046-A656	CFAD-047-0040	0B30
0B40	CFAD-048-3B35	CFAD-049-0008	CFAD-050-D1CB	CFAD-051-75AD	CFAD-052-8B4E	CFAD-053-C5C6	CFAD-054-755B	CFAD-055-D052	0B40
0B50	CFAD-056-0008	CFAD-057-B04A	CFAD-058-5AE2	CFAD-059-C5EE	CFAD-060-F5DE	CFAD-061-0002	CFAD-062-8B6E	CFAD-063-1E8D	0B50
0B60	CFAD-064-E070	CFAD-065-0008	CFAD-066-2002	CFAD-067-755B	CFAD-068-D06C	CFAD-069-0008	CFAD-070-9564	CFAD-071-E07F	0B60
0B70	CFAD-072-8178	CFAD-073-D5F9	CFAD-074-F5F9	CFAD-075-AA74	CFAD-077-CE7D	CFAD-078-A824	CFAD-079-A820	CFAD-080-F5A9	0B70
0B80	CFAD-082-5CC2	CFAD-083-D5CA	CFAD-084-D043	CFAD-085-0FF3	CFAD-086-C4C2	CFAD-087-B2F8	CFAD-088-2EFF	CFAD-089-E496	0B80
0B90	CFAD-090-F499	CFAD-091-2507	CFAD-092-8B86	CFAD-093-F493	CFAD-094-ED1F	CFAD-097-2C07	CFAD-098-B1A0	CFAD-099-2080	0B90
0BA0	CFAD-100-1E8D	CFAD-101-27D3	CFAD-102-2507	CFAD-103-8B86	CFAD-104-2A13	CFAD-106-2C13	CFAD-107-A656	CFAD-108-5CC2	0BA0
0BB0	CFAD-109-2A07	CFAD-110-2E1B	CFAD-111-E4BB	CFAD-112-F4BD	CFAD-113-8BC2	CFAD-114-F4B9	CFAD-115-1E8D	CFAD-116-27C3	0BB0
0BC0	CFAD-117-2C80	CFAD-118-2C25	CFAD-119-CE48	CFAD-120-1C7D	CFAD-121-B1A6	CFAD-123-FD4F	CFAD-124-2E07	CFAD-125-1E85	0BC0
0BD0	CFAD-126-896A	CFAD-127-817A	CFAD-128-2C07	CFAD-129-FD5B	CFAD-130-AA76	CFAD-131-2BE3	CFAD-132-89FC	5007	0BD0
0BE0	CICY-055-AA20	DCLE-034-FDEA	DCLE-035-B09C	DCLE-032-EDE2	DCLE-033-8172	DCLE-036-2B40	DCLE-037-2645	DCLE-038-3613	0BE0
0BF0	DCLE-039-BC86	DPTS-043-2482	DPTS-044-5449	DPTS-045-C4FC	DPTS-046-5460	DPTS-047-8D22	DPTS-048-9816	DPTS-049-54C9	0BF0

	0	2	4	6	8	A	C	E	0C--
0C00	DPTS-050-8F7A	DPTS-051-DE87	DPTS-052-3F45	DPTS-053-0C4B	DPTS-054-C4BF	DPTS-055-0C3B	DPTS-056-C4D7	DPTS-060-DE92	0C00
0C10	DPTS-061-1F45	DPTS-062-1D00	DPTS-063-1F97	DPTS-064-DF5F	DPTS-065-D85D	DPTS-066-F85C	DPTS-070-5CE9	DPTS-071-CC27	0C10
0C20	DPTS-072-0EE9	DPTS-073-F4AE	DPTS-077-A37E	DPTS-079-6EE3	DPTS-080-E0DD	DPTS-081-0E39	DPTS-085-F4B7	DPTS-087-2E05	0C20
0C30	DPTS-088-4DCF	DPTS-089-E83B	DPTS-090-8CF0	DPTS-092-E85C	DPTS-093-A398	DPTS-098-2D02	DPTS-099-8D6E	DPTS-104-F549	0C30
0C40	DPTS-105-5FB9	DPTS-106-E247	DPTS-107-1823	DPTS-108-7B38	DPTS-109-2E43	DPTS-110-5006	DPTS-111-C551	DPTS-112-C4D5	0C40
0C50	DPTS-113-3445	DPTS-114-8D4C	DPTS-115-8D56	DPTS-121-DEDE	DPTS-125-2EC3	DPTS-126-8D6E	DPTS-131-3F85	DPTS-132-2E23	0C50
0C60	DPTS-137-1547	DPTS-138-F268	DPTS-139-3543	DPTS-140-8D68	DPTS-141-5E29	DPTS-142-B404	DPTS-143-52E9	DPTS-144-1485	0C60
0C70	DPTS-146-C174	DPTS-147-3485	DPTS-148-54F9	DPTS-149-8178	DPTS-150-A030	CCOM-187-2BB1	CCOM-188-5FA0	CCOM-189-128E	0C70
0C80	DPTT-026-1548	DPTT-027-B448	DPTT-028-DA0F	DPTT-029-CA0E	DPTT-030-E40F	DPTT-031-2085	DPTT-032-8C90	DPTT-033-2007	0C80
0C90	DPTT-034-7068	DPTT-035-C116	DPTT-036-3485	DPTT-037-54F9	DPTT-038-7A68	DPTT-039-5068	DPTT-040-5260	DPTT-041-A00E	0C90
0CA0	DPTT-046-5E29	DPTT-047-CEB1	DPTT-049-DA39	DPTT-050-2E15	DPTT-051-54F9	DPTT-052-1F85	DPTT-053-F246	DPTT-054-A01E	0CA0
0CB0	DPTT-056-8F00	DPTT-057-5F39	DPTT-058-6664	DPTT-059-726A	DPTT-061-B448	DPTT-062-7460	DPTT-063-54F9	DPTT-064-E243	0CB0
0CC0	DPTT-065-CD08	DPTT-066-F22F	DPTT-067-3E15	DPTT-068-AC30	DPTT-006-2482	DPTT-007-8F7A	DPTT-008-D15B	DPTT-009-C421	0CC0
0CD0	DPTT-010-DED4	DPTT-011-810C	DPTT-013-2E23	DPTT-014-2F07	DPTT-015-A01E	DPTT-020-0D08	DPTT-021-5FB9	DPTT-023-E401	0CD0
0CE0	DPTT-024-AA19	DPTT-025-AECA	CLOG-118-B3F6	CLOG-119-6F15	CLOG-120-6E05	CLOG-121-6735	CLOG-122-6625	CLOG-123-A844	0CE0
0CF0	DPTR-007-EB74	DPTR-008-2D20	DPTR-009-2813	DPTR-010-2985	DPTR-011-293B	DPTR-012-6A99	DPTR-014-2C75	DPTR-018-5D38	0CF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	0C-- 0D--
0D00	DPTR-019-5DB9	DPTR-020-C48D	DPTR-021-5BC0	DPTR-022-0BFF	DPTR-023-C48C	DPTR-024-3F83	DPTR-025-4F8F	DPTR-026-2D04	0D00
0D10	DPTR-027-287B	DPTR-028-4BBF	DPTR-029-28AD	DPTR-030-F49B	DPTR-031-6893	DPTR-032-282B	DPTR-033-D42B	DPTR-034-5006	0D10
0D20	DPTR-035-C4A9	DPTR-036-2AFF	DPTR-037-C485	DPTR-038-8CFE	DPTR-039-C565	DPTR-040-2AFF	DPTR-041-C4B7	DPTR-042-3445	0D20
0D30	DPTR-043-2B05	DPTR-044-8D0C	DPTR-045-3445	DPTR-046-EABE	DPTR-047-3E23	DPTR-048-3F15	DPTR-049-1545	DPTR-050-2D05	0D30
0D40	DPTR-051-5DC0	DPTR-052-CD46	DPTR-053-1F83	DPTR-054-3D02	DPTR-055-1F23	DPTR-056-D456	DPTR-061-C555	DPTR-062-E554	0D40
0D50	DPTR-063-1445	DPTR-064-8D56	DPTR-065-1545	DPTR-066-3E83	DPTR-067-3485	DPTR-068-D568	DPTR-069-8FBA	DPTR-070-1E83	0D50
0D60	DPTR-071-DA6A	DPTR-072-8178	DPTR-073-B404	DPTR-074-95CA	DPTR-075-8FBE	DPTR-076-B406	DPTR-077-AB20	DPTR-082-C572	0D60
0D70	DPTR-083-15E5	DPTR-084-3485	DPTR-085-3E83	DPTR-086-D55D	DPTR-087-F269	DPTR-088-DA7E	DPTR-089-1485	DPTR-090-8C60	0D70
0D80	CDMD-150-A520	CDMD-106-571A	CDMD-107-A31B	CDMD-108-2FFF	CDMD-109-F483	CDMD-110-5004	CDMD-111-5224	CDMD-112-5EF9	0D80
0D90	CDMD-113-2B07	CDMD-114-E597	CDMD-115-2040	CDMD-116-0004	CDMD-117-77AF	CDMD-118-5730	CDMD-119-D1A1	CDMD-120-7718	0D90
0DA0	CDMD-121-7B38	CDMD-122-2FFF	CDMD-123-C498	CDMD-124-5006	CDMD-125-5226	CDMD-126-D1C3	CDMD-127-577B	CDMD-128-7710	0DA0
0DB0	CDMD-129-77AF	CDMD-130-C181	CDMD-131-E5C2	CDMD-132-B12E	CDMD-133-0B9D	CDMD-134-E0C2	CDMD-135-4086	CDMD-136-1002	0DB0
0DC0	CDMD-137-B3CA	CDMD-138-54F9	CDMD-139-8174	CDMD-140-27C3	CDMD-141-C94C	CDMD-142-176B	CDMD-143-EF51	CDMD-144-3713	0DC0
0DD0	CDMD-145-5F10	CDMD-146-6F75	CDMD-147-7F10	CDMD-148-2A07	CDMD-149-AA76	CDVD-036-AEF6	CDVD-037-3002	CDVD-038-79FD	0DD0
0DE0	CDVD-039-78ED	CDVD-040-757D	CDVD-041-746D	CDVD-042-F5DB	CDVD-043-AEF6	CDVD-044-1002	CDVD-045-8DDE	DCLC-014-C176	0DE0
0DF0	DCLC-015-7602	DCLC-016-2D40	DCLC-017-8DF8	DCLC-018-0D40	DCLC-019-2B80	DCLC-020-4F7F	DCLC-021-2B84	DCLC-022-2BC4	0DF0

	0	2	4	6	8	A	C	E	OE--
OE00	DCLC-023-C9DD	DCLC-024-FDA1	DCLC-025-262B	DCLC-026-E600	DCLC-027-05C3	DCLC-028-F094	DCLC-029-F514	DCLC-030-2323	OE00
OE10	DCLC-031-2207	DCLC-032-8E3A	DCLC-033-DD9F	DCLC-034-2B00	DCLC-035-F21F	DCLC-036-C3CB	DCLC-037-ADF6	DCLC-038-A154	OE10
OE20	DCLC-039-5F4F	DCLC-040-FC9F	DCLC-041-2B84	DCLC-042-E61F	DCLC-043-FDAE	DCLC-044-262B	DCLC-045-FDA7	DCLC-046-2B80	OE20
OE30	DCLC-047-F21F	DCLC-048-E149	DCLC-049-F149	DCLC-050-2307	DCLC-051-5429	DCLC-052-72A2	DCLC-053-2B00	DCLC-054-2400	OE30
OE40	DCLC-055-5EA2	DCLC-056-2A07	DCLC-057-A01E	DCLC-058-9C6A	DCLC-059-C3D8	DCLC-060-A7D6	DCLC-061-F537	DCLC-062-5449	OE40
OE50	DCLC-063-C4B6	DCLC-064-2B00	DCLC-065-810C	DCLC-066-2B84	DCLC-067-2307	DCLC-068-A7B6	DCLC-069-2BC0	DCLC-070-0D10	OE50
OE60	DCLC-071-E61F	DCLC-072-CDE9	DCLC-073-262B	DCLC-074-CDE0	DCLC-075-4F4F	DCLC-076-5F4F	DCLC-077-FC9F	DCLC-078-6471	OE60
OE70	DCLC-079-C49E	DCLC-080-054B	DCLC-081-C4D7	DCLC-082-2BC2	DCLC-083-9E86	CBIN-076-B3F6	CBIN-077-B460	CBIN-078-AD46	OE70
OE80	BWRP-016-F790	BWRP-018-96B4	BWRP-024-8F64	BWRP-030-904A	BWRP-032-904A	BWRP-040-5007	BWRP-042-904A	BWRP-046-B490	OE80
OE90	BWRP-083-2A07	BWRP-091-E1BC	BWRP-093-2440	BWRP-094-78F2	BWRP-095-2400	BWRP-096-56F2	BWRP-097-16C5	BWRP-098-371B	OE90
OEA0	BWRP-099-4660	BWRP-100-06BD	BWRP-101-E0AC	BWRP-102-064B	BWRP-103-F0AC	BWRP-104-F339	BWRP-106-08CD	BWRP-114-C4BA	OEA0
OEB0	BWRP-115-092B	BWRP-116-C4BA	BWRP-117-E23B	BWRP-118-6886	BWRP-119-4452	BWRP-120-AA20	BWRP-121-8176	BWRP-122-AA20	OEB0
OEC0			BCHK-321-0000	BCHK-321-0000	CICY-043-814C	CICY-044-55E5	CICY-045-61EB	CICY-046-60AD	OEC0
OED0	CICY-047-6FAD	CICY-048-55ED	CICY-049-63EB	CICY-050-62AD	CICY-051-67AD	CICY-052-6F75	CICY-053-AB77	CICY-054-B07D	OED0
OEE0	CICY-267-83B4	CICY-268-8136	CICY-269-AE6E	CICY-270-AE68	CICY-272-83B4	CICY-273-83B4	CICY-278-83B4	CICY-279-83B4	OEE0
OEF0	CICY-280-934C	CICY-281-93BA	CICY-282-934E	CICY-283-93B8	CICY-284-97A8	CICY-285-97A8	CICY-287-83B4	CICY-288-83B4	OEF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	OE-- OF--
OF00	DPTT-071-FE84	DPTT-072-3E13	DPTT-073-EE8A	DPTT-074-3E23	DPTT-075-3F13	DPTT-076-CB0E	DPTT-077-3E23	DPTT-078-5E3F	OF00
OF10	DPTT-079-1393	DPTT-080-F099	DPTT-081-3E23	DPTT-082-3F15	DPTT-083-3E43	DPTT-084-0D0A	DPTT-085-128E	BDIA-423-EE9E	OF10
OF20	BDIA-424-1212	BDIA-425-5EFF	BDIA-426-C4A4	BDIA-427-2800	BDIA-428-2020	BDIA-432-9324	CDMD-203-5773	CDMD-204-0769	OF20
OF30	CDMD-205-F485	CDMD-206-A520	CDMD-207-F73C	CDMD-208-0719	CDMD-209-F4BD	CDMD-210-143D	CDMD-211-128E	5007	OF30
OF40	DCLA-011-8938	DCLA-012-894E	DCLA-013-894E	DCLA-014-5C09	DCLA-018-C4D0	DCLA-020-C7CF	DCLA-041-810C	DCLA-042-AED0	OF40
OF50	DCLA-044-001B	DCLA-045-C4D7	DCLA-047-95BE	DCLA-049-2486	DCLA-050-D8CF	DCLA-058-2480	DCLA-059-E9CF	DCLA-064-5632	OF50
OF60	DCLA-065-C64C	DCLA-066-AECA	ERRQ-086-4642	ERRQ-087-3725	ERRQ-088-2402	ERRQ-089-C4F4	ERRQ-074-4642	ERRQ-075-3715	OF60
OF70	ERRQ-076-F768	ERRQ-077-2406	ERRQ-079-3743	ERRQ-080-5060	ERRQ-081-1773	ERRQ-082-5468	ERRQ-083-5E6A	ERRQ-084-128E	OF70
OF80	FILE-325-6CC3	FILE-326-6CC3	FILE-327-6CC3	FILE-328-4BEF	FILE-334-49CF	FILE-336-4EEF	FILE-337-4DEF	FILE-338-128E	OF80
OF90	FILE-288-2D95	FILE-289-0E08	FILE-290-2100	FILE-295-CCAB	FILE-296-2D1B	FILE-297-DCAB	FILE-298-2D1B	FILE-299-ECAB	OF90
OFA0	FILE-300-2D1B	FILE-301-FCAB	FILE-305-128E	FILE-316-5C32	FILE-317-0E08	FILE-318-2E07	FILE-319-C90B	FILE-321-D90B	OFA0
OFB0	FILE-323-2C15	FILE-324-891F	DCOM-073-C838	DCOM-074-2F04	DCOM-075-128E	DCOM-006-C174	DCOM-007-2683	DCOM-008-F047	OFB0
OFC0	DCOM-009-221E	DCOM-010-C7E9	DCOM-011-2010	DCOM-012-76D2	DCOM-013-021E	DCOM-014-3413	DCOM-015-1683	DCOM-025-F75B	OFC0
OFD0	DCOM-027-E757	DCOM-028-0D08	DCOM-029-128E	DCOM-034-0F20	DCOM-040-128E	DCOM-030-E761	DCOM-032-0B20	DCOM-033-128E	OFD0
OFE0	DCOM-041-0F24	DCOM-042-C866	DCOM-043-1C83	DCOM-044-128E	DCOM-045-021E	DCOM-046-1683	DCOM-047-D575	DCOM-057-F77B	OFE0
OFF0	DCOM-059-E777	DCOM-060-2D88	DCOM-061-128E	DCOM-066-2F30	DCOM-072-128E	DCOM-062-E735	DCOM-064-2B30	DCOM-065-128E	OFF0

	0	2	4	6	8	A	C	E	10--
1000	BMCK-032-C693	BMCK-036-D695	BMCK-040-E693	BMCK-044-F697	BMCK-045-C29B	BMCK-058-021A	BMCK-060-021C	BMCK-061-021E	1000
1010	BMCK-063-9C34	BMCK-065-98F8	BMCK-067-96AC	BMCK-071-2D02	BMCK-072-B22E	BMCK-073-2D40	BMCK-074-A4D8	5007	1010
1020	DCLB-245-8BF2	DCLB-247-9912	DCLB-248-9912	DCLB-253-A59A	BMCK-006-7612	BMCK-007-566F	BMCK-008-577F	BMCK-009-7672	1020
1030	BMCK-010-2610	BMCK-011-A281	BMCK-012-2400	BMCK-018-2135	BMCK-019-B368	CMLT-169-A752	CMLT-170-3002	CMLT-171-A378	1030
1040	DCLB-231-8DEE	CICY-061-9CFA	CICY-058-834A	CICY-059-C043	CICY-060-9023	BWRP-140-2BA5	BWRP-142-2607	BWRP-146-8F78	1040
1050	BWRP-147-64B5	BWRP-148-2543	BWRP-149-2EC3	BWRP-150-8761	FILX-080-3E00	FILX-081-2C25	FILX-082-7C52	FILX-083-8714	1050
1060	BWRP-152-8D68	BWRP-156-B01E	BWRP-157-B01E	BWRP-162-8792	FILE-149-AEEC	FILE-150-3E00	FILE-156-2490	FILE-157-0E08	1060
1070	FILE-158-2302	FILE-159-2304	FILE-160-3110	FILE-161-3D29	FILE-162-4FDF	FILE-163-43DF	FILE-164-8678	FILE-165-5FCF	1070
1080	FILE-166-1CFF	FILE-167-C4FC	FILE-168-4FCF	FILE-169-5FCF	FILE-170-C4FC	FILE-171-866C	FILE-172-866C	FILE-173-5BCF	1080
1090	FILE-174-867A	FILE-175-5ECF	FILE-176-C4FC	FILE-177-2310	FILE-178-2390	FILE-179-2D13	FILE-180-4FDF	FILE-181-2D25	1090
10A0	FILE-182-43DF	FILE-183-3110	FILE-184-8670	FILE-185-2D1F	FILE-186-4FDF	FILE-187-866C	FILE-188-8670	FILE-189-5BCF	10A0
10B0	FILE-190-2D85	FILE-191-3D23	FILE-192-867A	FILE-193-3D73	FILE-194-8676	FILE-195-3114	FILE-196-866C	FILE-197-2DA3	10B0
10C0	FILE-198-8676	FILE-199-1D00	FILE-200-CAFD	FILE-201-2380	FILE-202-3114	FILE-203-866C	FILE-204-3D47	FILE-205-8676	10C0
10D0	FILE-206-1D00	FILE-207-2390	FILE-208-3114	FILE-209-2308	FILE-210-2D75	FILE-211-3DA3	FILE-212-8676	FILE-213-1D00	10D0
10E0	FILE-214-3114	FILE-215-2380	FILE-216-2308	FILE-217-1D45	FILE-218-8676	FILE-219-1D00	FILE-220-3114	FILE-221-2308	10E0
10F0	FILE-222-2D55	FILE-223-3DB3	FILE-224-8676	FILE-225-1D00	FILE-226-2480	FILE-227-9108	FILE-228-2100	FILE-229-1E00	10F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	10-- 11--
1100	FILE-230-0E08	FILE-231-2480	FILE-232-3C83	FILE-233-8504	FILE-234-21C0	FILE-235-3E00	FILE-236-0E08	FILE-237-0E04	1100
1110	FILE-238-2D43	FILE-239-4EDF	FILE-240-5CC2	FILE-241-2A07	FILE-242-2490	FILE-243-5E42	FILE-244-831A	5007	1110
1120	CICY-217-AA68	CICY-218-83B4	CICY-219-AA24	CICY-220-AA30	CICY-221-8898	CICY-222-8A5E	CICY-223-AC88	CICY-224-AC88	1120
1130	FINT-050-2A07	FINT-051-2863	FINT-052-5EA0	FINT-053-0D08	FINT-054-526F	FINT-055-566B	FINT-056-2707	FINT-057-C9C2	1130
1140	FINT-058-3F85	FINT-059-2490	FINT-060-78A2	FINT-061-0D04	FINT-062-2400	FINT-063-50A2	FINT-064-5242	FINT-065-128E	1140
1150	EXFR-007-2455	EXFR-008-2515	EXFR-009-3593	EXFR-010-2045	EXFR-011-21A5	EXFR-012-22A3	EXFR-013-2363	EXFR-014-4852	1150
1160	EXFR-015-4A86	EXFR-016-4C86	EXFR-017-4E86	EXFR-018-2613	EXFR-019-5700	EXFR-020-C76F	EXFR-021-6965	EXFR-022-D773	1160
1170	EXFR-023-6B65	EXFR-024-E777	EXFR-025-6D65	EXFR-026-F77B	EXFR-027-6F65	EXFR-028-C37F	EXFR-029-6865	EXFR-030-D303	1170
1180	EXFR-031-6A65	EXFR-032-E307	EXFR-033-6C65	EXFR-034-F30B	EXFR-035-6E65	EXFR-036-6663	EXFR-037-23FF	EXFR-038-C49F	1180
1190	EXFR-040-F41A	EXFR-042-21AB	EXFR-043-F498	EXFR-044-201D	EXFR-045-9168	EXFR-047-319B	EXFR-048-9168	EXFR-050-7940	1190
11A0	EXFR-051-6444	EXFR-052-7840	EXFR-053-6444	EXFR-054-7D40	EXFR-055-6444	EXFR-056-7F40	EXFR-057-6444	EXFR-058-144B	11A0
11B0	EXFR-059-D03C	EXFR-060-5899	EXFR-061-5AB9	EXFR-062-5CD9	EXFR-063-5EF9	EXFR-064-919E	EXFR-065-22FF	EXFR-066-C4CF	11B0
11C0	EXFR-068-F44A	EXFR-070-313D	EXFR-071-C048	EXFR-072-20FD	EXFR-073-915C	EXFR-075-213F	EXFR-076-915C	EXFR-078-148B	11C0
11D0	EXFR-079-CC60	EXFR-081-F45A	EXFR-083-211B	EXFR-084-35AD	EXFR-085-915A	EXFR-087-21A7	EXFR-088-2507	EXFR-089-915A	11D0
11E0	EXFR-090-F475	EXFR-091-52E2	EXFR-092-8F64	EXFR-093-51A2	EXFR-094-8A68	EXFR-095-54B2	EXFR-096-3659	EXFR-097-2845	11E0
11F0	EXFR-098-2907	EXFR-099-99DE	EXFR-100-2466	EXFR-104-E67E	EXFR-105-1625	EXFR-106-2400	EXFR-107-AB20	EXFR-108-2406	11F0

	0	2	4	6	8	A	C	E	12--
1200	EXFR-109-8F6C	EXFR-110-8A68	EXFR-111-5FB2	EXFR-112-FB0A	EXFR-113-2643	EXFR-114-FD8E	EXFR-115-3D15	EXFR-116-C992	1200
1210	EXFR-117-3D83	EXFR-118-2E23	EXFR-119-3485	EXFR-120-A925	EXFR-121-2E07	EXFR-122-F822	EXFR-123-E523	EXFR-124-3445	1210
1220	EXFR-125-1545	EXFR-126-B3E8	EXFR-127-D529	EXFR-128-3E83	EXFR-129-D22E	EXFR-130-56F2	EXFR-131-8D6E	EXFR-132-51A2	1220
1230	EXFR-133-2007	EXFR-134-D83A	EXFR-135-52F2	EXFR-136-613B	EXFR-137-602D	EXFR-138-C844	EXFR-139-3E43	EXFR-140-C3C4	1230
1240	EXFR-141-2820	EXFR-142-8D6A	EXFR-143-C3C3	EXFR-144-8D5A	CCOM-182-5682	CCOM-183-427F	CCOM-184-128E	FILX-385-3F85	1240
1250	FILX-386-7E62	FILX-387-FFD6	FILX-388-51EF	FILX-389-51CF	FILX-390-2D07	FILX-391-4BDF	FILX-392-2D15	FILX-393-7CE3	1250
1260	FILX-394-F4E8	FILX-395-2C1B	FILX-396-1CFF	FILX-397-5FFD	FILX-398-4EDF	FILX-399-4BDF	FILX-400-2D07	FILX-401-4BDF	1260
1270	FILX-402-4EEF	FILX-403-2E45	FILX-404-4BEF	FILX-405-4BDF	FILX-406-4EFF	FILX-407-2E25	FILX-408-4BEF	FILX-409-4BDF	1270
1280	FILX-410-0CFF	FILX-411-C496	FILX-412-2F43	FILX-413-4EFF	FILX-414-2EC3	FILX-415-DFA6	FILX-416-2D15	FILX-417-4BDF	1280
1290	FILX-418-5E42	FILX-419-1D00	FILX-420-0222	FILX-421-4ECF	FILX-422-2F85	FILX-423-4BFF	FILX-424-4BDF	FILX-425-4EEF	1290
12A0	FILX-426-2D13	FILX-427-4BDF	FILX-428-2E83	FILX-429-2F45	FILX-430-CDBB	FILX-431-F9B1	FILX-432-4886	FILX-433-C4BA	12A0
12B0	FILX-434-DDB8	FILX-435-DA36	FILX-436-8640	FILX-437-85FE	FILX-438-2F07	FILX-439-4A52	FILX-440-7A62	FILX-441-4DAF	12B0
12C0	FILX-443-4FAF	FILX-445-1E00	FILX-446-1D00	FILX-447-2E08	FILX-448-5C32	FILX-449-C3DE	FILX-450-0EC3	FILX-451-F0D3	12C0
12D0	FILX-452-A7D8	FILX-453-0F85	FILX-454-C4D0	FILX-455-2C07	FILX-456-7C32	FILX-457-0212	FILX-458-A7F2	FILX-459-2B43	12D0
12E0	FILX-460-7CA8	FILX-461-DC70	FILX-462-7A32	FILX-463-C9EC	FILX-464-3F85	FILX-465-0D04	FILX-466-2410	FILX-467-A030	12E0
12F0	FILX-468-7EA0	FILX-469-2D08	FILX-470-8602	FILX-471-5EEF	FILX-472-1EC7	FILX-473-C4D0	FILX-474-5E42	FILX-475-96DC	12F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	12-- 13--
1300	BCHK-105-A4E6	BCHK-069-4252	BCHK-070-4C26	BCHK-071-4AC0	BCHK-072-2DEB	BCHK-073-8258	BCHK-074-6CC4	BCHK-083-DC0A	1300
1310	BCHK-092-42A6	BCHK-093-C481	BCHK-094-2CE3	BCHK-095-2DC5	BCHK-096-3D63	BCHK-097-C99F	BCHK-098-2820	BCHK-100-8258	1310
1320	BCHK-101-6AC0	BCHK-102-932C	BCHK-011-2404	BCHK-051-4632	BCHK-052-6662	BCHK-053-FA82	BCHK-054-21C5	BCHK-055-20E3	1320
1330	BCHK-056-6004	BCHK-057-4208	BCHK-058-2A07	BCHK-059-53BD	BCHK-060-C4BC	BCHK-061-2B1B	BCHK-062-6BB3	BCHK-063-42A0	1330
1340	BCHK-064-6208	BCHK-065-0610	BCHK-066-511B	BCHK-067-2723	BCHK-068-9A8C	5007	CSAS-006-3443	CSAS-007-1082	1340
1350	CSAS-008-2E15	CSAS-009-573A	CSAS-010-576D	CSAS-011-0661	CSAS-012-E0F3	CSAS-013-00C4	CSAS-014-F366	CSAS-015-07FB	1350
1360	CSAS-016-F0E7	CSAS-017-C5F5	CSAS-018-3042	CSAS-019-C07C	CSAS-020-D07B	CSAS-021-148B	CSAS-022-57FB	CSAS-023-5710	1360
1370	CSAS-024-9354	CSAS-025-9BDC	CSAS-026-1002	CSAS-027-2040	CSAS-028-9366	CSAS-029-1002	CSAS-030-577B	CSAS-031-77FF	1370
1380	CSAS-032-C905	CSAS-033-272B	CSAS-034-D188	CSAS-035-271B	CSAS-036-27AB	CSAS-037-F035	CSAS-038-771A	CSAS-039-55B9	1380
1390	CSAS-040-E0E8	CSAS-041-F0A6	CSAS-042-2707	CSAS-043-2EF3	CSAS-044-D02A	CSAS-045-2F07	CSAS-046-93AC	CSAS-047-7F1A	1390
13A0	CSAS-048-75E3	CSAS-049-E0EB	CSAS-050-F095	CSAS-051-573A	CSAS-052-D01B	CSAS-053-5F10	CSAS-054-7F7F	CSAS-055-F01E	13A0
13B0	CSAS-056-5006	CSAS-057-93A0	CSAS-058-5006	CSAS-059-938E	CSAS-060-1413	CSAS-061-3002	CSAS-062-9350	CSAS-063-573A	13B0
13C0	CSAS-064-F5D1	CSAS-065-7A7F	CSAS-066-1A9F	CSAS-067-C4F9	CSAS-068-0002	CSAS-069-1415	CSAS-070-2A07	CSAS-071-93F8	13C0
13D0	CSAS-072-E5D9	CSAS-073-7A7F	CSAS-074-C4F9	CSAS-075-93CA	CSAS-076-7A7F	CSAS-077-F5E1	CSAS-078-E445	CSAS-079-93CA	13D0
13E0	CSAS-080-142D	CSAS-081-93F8	CSAS-082-77AF	CSAS-083-93F6	CSAS-084-9372	CSAS-085-F0FD	CSAS-086-C5BF	CSAS-087-573A	13E0
13F0	CSAS-088-F5E5	CSAS-089-77AF	CSAS-090-C4F9	CSAS-091-2002	CSAS-092-25FF	CSAS-093-F0EC	CSAS-094-C1E9	CSAS-095-C591	13F0

	0	2	4	6	8	A	C	E	14--
1400	CSAS-096-F589	CSAS-097-E58B	CSAS-098-D1D1	CSAS-099-AA74	CSAS-100-F03C	CSAS-101-D18F	CSAS-102-A824	CSAS-103-A820	1400
1410	CSAS-104-F583	CSAS-105-F047	CSAS-106-5BB5	CSAS-107-61B9	CSAS-108-60AD	CSAS-109-5710	CSAS-110-57FB	CSAS-111-3002	1410
1420	CSAS-112-577D	CSAS-113-171B	CSAS-114-7AFF	CSAS-115-67A3	CSAS-116-771A	CSAS-117-5BA9	CSAS-118-C4BB	CSAS-119-5710	1420
1430	CSAS-120-757F	CSAS-121-751A	CSAS-122-2507	CSAS-123-2AFF	CSAS-124-C4AE	CSAS-125-F447	CSAS-126-2C75	CSAS-127-DD43	1430
1440	CSAS-128-AA76	CSAS-129-2BA3	CSAS-130-89FC	CSAS-131-D1CD	CSAS-132-2040	CSAS-133-9402	CSAS-134-0040	CSAS-135-9402	1440
1450	CSAS-136-F007	CSAS-137-9414	CDMD-184-5C79	CDMD-185-4286	CDMD-186-5D30	CDMD-187-5F6A	CDMD-188-7DFF	CDMD-189-7D3A	1450
1460	CDMD-190-C359	CDMD-191-7BAF	CDMD-192-0002	CDMD-193-C5EC	CDMD-194-0B9F	CDMD-195-2002	CDMD-196-D1F2	CDMD-197-E0D4	1460
1470	CDMD-198-B13E	CDMD-199-F0D4	CDMD-200-5710	CDMD-201-47BB	CDMD-202-B134	BMCK-092-2007	BMCK-093-2400	BMCK-094-5E3F	1470
1480	BMCK-095-C4A7	BMCK-096-1212	BMCK-103-5672	BMCK-104-2440	BMCK-105-7672	BMCK-106-2400	BMCK-107-2185	BMCK-108-420F	1480
1490	BMCK-109-7618	BMCK-110-7318	BMCK-111-5710	BMCK-112-271B	BMCK-113-7718	BMCK-114-5672	BMCK-115-16C5	BMCK-116-1713	1490
14A0	BMCK-117-6666	BMCK-118-7618	BMCK-119-9248	BMCK-120-2404	BMCK-121-2233	BMCK-122-1F2E	BMCK-123-2171	BMCK-124-2B07	14A0
14B0	BMCK-125-7A00	BMCK-126-4632	BMCK-127-261D	BMCK-128-271D	BMCK-129-6662	BMCK-130-17AB	BMCK-131-2185	BMCK-132-88BC	14B0
14C0	BMCK-133-9A8E	CBIN-005-8270	CBIN-006-AD54	EPCH-034-C86D	EPCH-035-2462	EPCH-036-3643	EPCH-037-2402	EPCH-038-1C83	14C0
14D0	EPCH-042-950C	EPCH-032-99B4	EPCH-010-1F00	EPCH-014-FA74	EPCH-016-6CF7	EPCH-017-EA53	EPCH-018-CE62	EPCH-019-DE53	14D0
14E0	EPCH-020-3C45	EPCH-021-DE66	EPCH-022-3C25	EPCH-023-E853	EPCH-024-CA46	EPCH-026-3C23	EPCH-027-2F40	EPCH-028-2462	14E0
14F0	EPCH-029-1643	EPCH-030-94CC	EPCH-044-EA52	EPCH-045-DA53	EPCH-046-0E49	EPCH-047-F4D2	EPCH-048-EE53	EPCH-049-EF84	14F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	14-- 15--
1500	EPCH-050-2C07	EPCH-051-99B0	EPCH-053-C808	EPCH-054-2D23	EPCH-055-3C83	EPCH-056-1C23	EPCH-057-3659	EPCH-058-5009	1500
1510	EPCH-059-C4A3	EPCH-060-3C53	EPCH-061-4A06	EPCH-062-2007	EPCH-063-21A5	EPCH-064-2002	EPCH-065-7B19	EPCH-066-7A09	1510
1520	EPCH-067-7AF2	EPCH-068-B3E8	EPCH-069-71A2	EPCH-070-72E2	EPCH-071-F72C	EPCH-072-99C4	EPCH-074-C82B	EPCH-075-CC42	1520
1530	EPCH-079-5CD9	EPCH-080-59C9	EPCH-081-C85E	EPCH-082-5809	EPCH-083-D83F	EPCH-084-5224	EPCH-085-282B	EPCH-086-58D5	1530
1540	EPCH-087-9BB0	EPCH-088-2B07	EPCH-089-2C45	EPCH-090-2DC3	EPCH-091-28F7	EPCH-092-5899	EPCH-093-54A0	EPCH-094-78A8	1540
1550	EPCH-095-74C8	EPCH-096-CD4C	EPCH-097-D94C	EPCH-098-5884	EPCH-099-3CE9	EPCH-100-2D07	EPCH-101-EE63	EPCH-102-1CA3	1550
1560	EPCH-103-9CC0	EPCH-104-9B4A	CFCO-103-F46F	CFCO-104-711B	CFCO-105-700B	CFCO-106-733B	CFCO-107-722B	CFCO-108-799B	1560
1570	CFCO-109-788B	CFCO-110-7FFB	CFCO-111-E079	CFCO-112-0002	CFCO-113-128E	DYPE-242-5AFF	DYPE-243-5F19	DYPE-244-11C5	1570
1580	DYPE-245-CF10	DYPE-246-2055	DYPE-247-21BD	DYPE-248-218B	DYPE-249-5F00	DYPE-250-4FFF	DYPE-251-0F20	DYPE-252-128E	1580
1590	DYPE-253-C483	DYPE-254-F098	DYPE-255-11CB	DYPE-256-C482	DYPE-257-E0B1	DYPE-258-CB20	DYPE-259-DB03	DYPE-260-EB2D	1590
15A0	DYPE-261-011F	DYPE-262-C4A9	DYPE-263-1FBD	DYPE-264-C48A	DYPE-265-1F7D	DYPE-266-C48A	DYPE-267-111B	DYPE-269-C482	15A0
15B0	DYPE-270-CB34	DYPE-271-EB39	DYPE-272-1FFD	DYPE-273-C48A	DYPE-274-114B	DYPE-275-C482	DCHN-127-2400	DCHN-128-2C75	15B0
15C0	DCHN-129-99F8	DCLB-110-6224	DCLB-111-6224	DCLB-112-6224	DCLB-113-9614	DCLB-043-1683	DCLB-044-3613	DCLB-045-3773	15C0
15D0	DCLB-046-5260	DCLB-047-D25A	DCLB-048-1643	DCLB-049-3363	DCLB-050-6224	DCLB-058-0249	DCLB-060-F4C3	DCLB-061-5038	15D0
15E0	DCLB-062-008B	DCLB-063-F0FA	DCLB-065-5119	DCLB-066-C4C4	DCLB-067-5030	DCLB-075-0049	DCLB-077-F4C5	DCLB-078-0183	15E0
15F0	DCLB-079-F0C4	DCLB-080-4206	DCLB-081-5038	DCLB-082-008B	DCLB-083-F0C5	DCLB-084-5119	DCLB-085-C4C4	DCLB-086-5049	15F0

	0	2	4	6	8	A	C	E	16--
1600	DCLB-087-6224	DCLB-088-5038	DCLB-089-0083	DCLB-090-F094	DCLB-091-C51F	DCLB-092-1537	DCLB-093-F031	DCLB-094-E033	1600
1610	DCLB-095-C035	DCLB-096-D033	DCLB-097-6224	DCLB-099-D639	DCLB-100-3763	DCLB-104-7260	DCLB-105-B002	DCLB-118-2405	1610
1620	DCLB-119-15F5	DCLB-120-6505	DCLB-121-5038	DCLB-122-4006	DCLB-123-C497	DCLB-125-D63D	DCLB-127-726A	DCLB-128-963E	1620
1630	DCLB-114-153B	DCLB-115-151B	DCLB-116-153B	DCLB-117-9620	DCLB-107-7212	DCLB-108-B002	DCLB-130-7212	DCLB-132-6226	1630
1640	DCLB-133-1343	DCLB-135-5230	DCLB-146-D659	DCLB-148-C54A	DCLB-149-1565	DCLB-174-1763	DCLB-175-756A	DCLB-189-6666	1640
1650	DCLB-190-F265	DCLB-191-5FF2	DCLB-192-7F60	DCLB-193-9666	DCLB-215-A041	DCLB-219-C160	DCLB-220-7602	DCLB-221-2D40	1650
1660	DCLB-223-1D00	DCLB-227-0244	DCLB-194-5F60	DCLB-195-6664	DCLB-196-42FF	DCLB-203-8721	CFCO-069-4E52	CFCO-071-48E6	1660
1670	CFCO-072-42E6	CFCO-073-40E6	CFCO-074-128E	CDVD-088-2002	CDVD-089-711B	CDVD-090-700B	CDVD-091-733B	CDVD-092-722B	1670
1680	CDVD-093-799B	CDVD-094-788B	CDVD-095-755B	CDVD-096-744B	CDVD-097-128E	CDVD-100-2002	CDVD-101-F581	CDVD-102-0008	1680
1690	CDVD-103-3002	CDVD-104-711B	CDVD-105-700B	CDVD-106-733B	CDVD-107-722B	CDVD-108-128E	CBIN-047-AE06	CBIN-048-88DC	1690
16A0		5007	FILT-086-50FF	FILT-087-0F83	FILT-088-F0BD	FILT-089-DB2E	FILT-090-8220	FILT-091-2F43	FILT-092-EB32	16A0
16B0	FILT-093-2F15	FILT-094-FB36	FILT-095-2F25	FILT-096-0680	FILT-097-2EC3	FILT-098-977E	FILT-099-2F07	FILT-100-C1B8	16B0
16C0	FILT-101-E5FE	FILT-102-2D45	FILT-103-4EDF	FILT-104-2E15	FILT-105-4BEF	FILT-106-2E43	FILT-107-4BFF	FILT-108-1D00	16C0
16D0	FILT-109-4EEF	FILT-110-FBD6	FILT-111-973C	FILT-112-C3FD	FILT-113-FD59	FILT-114-92F6	FILT-115-6EE6	FILT-116-6EE6	16D0
16E0	FILT-117-5CF8	FILT-118-4DCF	FILT-119-2C63	FILT-120-4FCF	FILT-121-5C52	FILT-122-2C07	FILT-123-F8EC	FILT-124-7C52	16E0
16F0	FILT-125-0E04	FILT-126-3D00	FILT-127-2C15	FILT-128-4BCF	FILT-129-6EE4	FILT-130-C3FB	FILT-131-0222	FILT-132-CA86	16F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	16-- 17--
1700	FILT-133-DF87	FILT-134-4BFF	FILT-135-870E	FILT-136-5ADF	FILT-137-C4C0	FILT-138-5C62	FILT-139-CA9F	FILT-140-C5C0	1700
1710	FILT-141-5BCF	FILT-142-6CD1	FILT-143-F098	FILT-144-928C	FILT-145-2D07	FILT-146-2C13	FILT-147-976E	FILT-148-5E52	1710
1720	FILT-149-EB31	FILT-150-CD32	FILT-151-2D1B	FILT-152-7C62	FILT-153-5DF9	FILT-154-0DAB	FILT-155-F0B5	FILT-156-9254	1720
1730	FILT-157-876A	FILT-158-859A	FILT-159-2D25	FILT-160-2C07	FILT-161-2F07	FILT-162-976E	FILT-163-FAC1	FILT-164-8640	1730
1740	FILT-165-2D07	FILT-166-2C45	FILT-167-DEEE	FILT-168-2E85	FILT-169-EEDD	FILT-170-FA9B	FILT-171-5ACF	FILT-172-C4E0	1740
1750	FILT-173-7C62	FILT-174-2EC3	FILT-175-D8F9	FILT-176-C8FB	FILT-177-2D07	FILT-178-2E15	FILT-179-2C15	FILT-180-9770	1750
1760	FILT-181-5CDB	FILT-182-1CF5	FILT-183-ED68	FILT-184-3D83	FILT-185-FD6C	FILT-186-3D23	FILT-187-1D35	FILT-188-2E07	1760
1770	FILT-189-7E62	FILT-190-2EE3	FILT-191-3C25	FILT-192-D8FA	FILT-193-3E13	FILT-194-7C52	FILT-195-2F07	FILT-196-F082	1770
1780	FILT-197-3E45	FILT-198-C986	FILT-199-3F85	FILT-200-1D04	FILT-201-C891	FILT-202-F993	FILT-203-4886	FILT-204-C493	1780
1790	FILT-205-3F45	FILT-206-2A07	FILT-207-0F85	FILT-208-C49A	FILT-209-EA24	FILT-210-2B43	FILT-211-3B15	FILT-212-5C42	1790
17A0	FILT-213-7EA8	FILT-214-7CA0	FILT-215-92BE	CDMD-047-A51C	CDMD-002-E1AD	CDMD-003-1445	CDMD-004-C130	CDMD-005-AE94	17A0
17B0	CDMD-006-5575	CDMD-007-77E3	CDMD-008-E0AE	CDMD-009-9812	CDMD-010-5730	CDMD-011-8F2C	CDMD-012-5710	CDMD-013-3415	17B0
17C0	CDMD-014-8F2C	CDMD-015-10CE	CDMD-016-55BD	CDMD-017-3B83	CDMD-018-3B15	CDMD-019-7AE2	CDMD-020-4CA6	CDMD-021-3B25	17C0
17D0	CDMD-022-7AF2	CDMD-023-5F3A	CDMD-024-5FFB	CDMD-025-7FAF	CDMD-026-7FCA	CDMD-027-5FF1	CDMD-028-C96C	CDMD-029-573A	17D0
17E0	CDMD-030-47F3	CDMD-031-77AF	CDMD-032-7FAA	CDMD-033-77CA	CDMD-034-57F5	CDMD-035-C95F	CDMD-036-7FAA	CDMD-037-55E5	17E0
17F0	CDMD-038-55FD	CDMD-039-7EF1	CDMD-040-2002	CDMD-041-71E9	CDMD-042-70A9	CDMD-043-4206	CDMD-045-F027	CDMD-046-8D82	17F0

	0	2	4	6	8	A	C	E	18--
1800	ECOL-086-2F08	ECOL-087-980A	ECOL-088-2F0C	ECOL-089-980A	ECOL-090-2F0A	ECOL-091-221B	ECOL-092-F491	ECOL-093-3C13	1800
1810	ECOL-094-921A	CCOM-056-70A2	CCOM-057-74B2	CCOM-058-7CC2	CCOM-059-E19E	CCOM-060-7892	CCOM-061-0020	CCOM-062-128E	1810
1820	FILX-276-5C84	FILX-277-2D3F	FILX-278-7EC0	FILX-279-86F2	FILX-266-988E	FILX-241-F930	FILX-242-4C52	FILX-243-7C62	1820
1830	FILX-244-7CE8	FILX-245-5C62	FILX-246-5EDF	FILX-247-1DA3	FILX-248-DD3C	FILX-249-3D83	FILX-250-7CE0	FILX-251-4C52	1830
1840	FILX-252-3F15	FILX-253-7CEA	FILX-254-1F1D	FILX-255-FBEF	FILX-256-5DE8	FILX-257-D5CF	FILX-258-7DB8	FILX-259-5886	1840
1850	FILX-260-C4FD	FILX-261-CB46	FILX-262-1FEB	FILX-263-FF44	FILX-264-FBA9	FILX-265-8634	FILX-229-1C25	FILX-230-7C52	1850
1860	FILX-231-5E32	FILX-232-1E45	FILX-233-7E32	FILX-234-2E07	FILX-235-2F25	FILX-236-3F43	FILX-237-FBAA	FILX-238-5DB8	1860
1870	FILX-239-7DE8	FILX-240-984E	FILX-225-3E00	FILX-226-5FFF	FILX-227-E0DD	FILX-228-85AC	FILX-267-CDA1	FILX-268-CB06	1870
1880	FILX-269-FF06	FILX-270-FB8F	FILX-271-8634	FILX-272-2F45	FILX-273-FBC7	FILX-274-F985	FILX-275-865A	FILX-288-2A07	1880
1890	FILX-289-2B25	FILX-290-3B43	FILX-291-5CA8	FILX-292-4CC6	FILX-293-C4A4	FILX-294-5CA0	FILX-295-3B17	FILX-296-5EAA	1890
18A0	FILX-297-6EC5	FILX-298-C4AF	FILX-299-2C13	FILX-300-2F45	FILX-301-CDC5	FILX-302-F9C0	FILX-303-98CA	FILX-307-5DE9	18A0
18B0	FILX-308-6BE3	FILX-309-F4B8	FILX-310-0EFF	FILX-311-C4A4	FILX-312-2CA3	FILX-313-7CF3	FILX-314-F4A4	FILX-315-924E	18B0
18C0	FILX-304-4886	FILX-305-C4AD	FILX-306-98CC	FILX-280-2C13	FILX-281-F9CC	FILX-282-2F07	FILX-283-3CA5	FILX-284-2D07	18C0
18D0	FILX-285-7C52	FILX-286-2EE3	FILX-287-92BA	DCLR-136-3415	DCLR-137-15C5	DCLR-138-1533	DCLR-140-81CA	ERCX-054-3B23	18D0
18E0	ERCX-055-9DCC	ERCX-056-9DC8	ERCX-057-98DE	ERCX-058-98EE	ERCX-059-9DCA	ERCX-060-98EE	ERCX-061-98EE	ERCX-065-F775	18E0
18F0	ERCX-066-2F80	ERCX-067-9DCC	ERCX-068-2880	ERCX-070-9DCC	DCLR-238-2343	DCLR-239-5612	DCLR-240-3613	DCLR-241-A19C	18F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	18-- 19--
1900	EPXF-041-2DAB	EPXF-042-2DAB	EPXF-043-2DAB	EPXF-044-2DAB	EPXF-045-2DAB	EPXF-046-2DAB	EPXF-047-9C74	EPXF-048-9C80	1900
1910	5007	ERDR-053-54E9	ERDR-054-C4A0	ERDR-055-5EB2	ERDR-056-5D69	ERDR-057-8A68	ERDR-058-3C85	ERDR-059-950E	1910
1920	ERDR-060-C3AE	ERDR-061-F72C	ERDR-062-2406	ERDR-063-CEAC	ERDR-064-3D23	ERDR-065-DDAF	ERDR-066-9F5C	ERDR-067-9812	1920
1930	ERDR-068-8A68	ERDR-069-CEBD	ERDR-070-8178	ERDR-071-2C75	ERDR-072-4452	ERDR-073-8D6A	ERDR-074-5460	ERDR-075-F747	1930
1940	ERDR-076-2D80	ERDR-077-0F30	ERDR-078-994A	ERDR-079-2D04	ERDR-080-CB30	ERDR-081-2FA3	ERDR-082-0E3B	ERDR-083-C4DC	1940
1950	ERDR-084-F758	ERDR-085-DDDA	ERDR-086-B43C	ERDR-087-8C58	ERDR-088-CFD5	ERDR-089-99B6	ERDR-090-0E4B	ERDR-091-F76F	1950
1960	ERDR-092-F0E9	ERDR-093-2D07	ERDR-094-CFDA	ERDR-095-94D4	ERDR-096-CFF3	ERDR-097-3D45	ERDR-098-9974	ERDR-099-F0FA	1960
1970	ERDR-100-DDEA	ERDR-101-1D45	ERDR-102-B43C	ERDR-103-3623	ERDR-104-A5E0	ERDR-105-2D07	ERDR-106-DDDA	ERDR-107-1B00	1970
1980	ERDR-108-6CF7	ERDR-109-EA34	ERDR-110-FA26	ERDR-111-EE34	ERDR-112-E835	ERDR-113-76F2	ERDR-114-CE1A	ERDR-115-DE18	1980
1990	ERDR-116-FA35	ERDR-117-0D04	ERDR-118-8F7A	ERDR-119-920A	ERDR-120-3C45	ERDR-121-DE1E	ERDR-122-3C25	ERDR-123-1C83	1990
19A0	ERDR-124-3C23	ERDR-125-B3E8	ERDR-126-9C05	ERDR-127-DA35	ERDR-128-EDAF	ERDR-129-2B40	ERDR-130-9504	ERDR-131-3D00	19A0
19B0	ERDR-133-2E13	ERDR-137-99B8	ERDR-138-2D85	ERDR-139-2E23	ERDR-140-B3E8	ERDR-141-B43C	ERDR-142-2A07	ERDR-143-C3C2	19B0
19C0	ERDR-144-9F5C	ERDR-145-8C60	ERDR-146-CC5D	ERDR-148-F753	ERDR-149-2445	ERDR-150-2513	ERDR-151-40A6	ERDR-152-31BB	19C0
19D0	ERDR-153-915A	ERDR-154-EE56	ERDR-158-36A9	ERDR-159-2855	ERDR-160-2915	ERDR-161-3983	ERDR-162-5E49	ERDR-164-F565	19D0
19E0	ERDR-168-2A35	ERDR-169-9DD4	ERDR-170-261B	ERDR-171-F4EA	ERDR-172-9DA6	ERDR-173-5884	ERDR-174-E471	ERDR-175-5884	19E0
19F0	ERDR-176-21FF	ERDR-177-C4E4	ERDR-178-9DA8	DCHN-122-2400	DCHN-123-2A07	DCHN-124-C3FE	DCHN-125-A7D6	DCHN-126-AA76	19F0

	0	2	4	6	8	A	C	E	1A--
1A00	ERDR-180-2B08	ERDR-181-9150	ERDR-182-2B0C	ERDR-183-9150	ERDR-184-2B0A	ERDR-185-9150	ALDP-037-26A3	ALDP-038-2783	1A00
1A10	ALDP-039-88BC	ALDP-040-88C4	ALDP-041-3643	ALDP-042-043D	ALDP-043-E0B3	ALDP-044-88BC	ALDP-045-24B3	ALDP-046-2577	1A10
1A20	ALDP-047-534A	ALDP-048-88CE	ALDP-049-D121	ALDP-050-D00D	ALDP-051-7543	ALDP-052-F520	ALDP-053-24F3	ALDP-054-F320	1A20
1A30	ALDP-055-9A28	ALDP-056-044B	ALDP-057-F0BD	ALDP-058-041B	ALDP-059-F09A	ALDP-060-36D9	ALDP-061-88C6	ALDP-062-3645	1A30
1A40	ALDP-085-042D	ALDP-086-E0CA	ALDP-087-3715	ALDP-088-047B	ALDP-089-9A5E	ALDP-090-043F	ALDP-091-E09A	ALDP-095-F0D4	1A40
1A50	ALDP-097-E66C	ALDP-098-C9ED	ALDP-100-042B	ALDP-101-F0DC	ALDP-102-E61B	ALDP-103-AF78	ALDP-105-041B	ALDP-115-F0E4	1A50
1A60	ALDP-117-3615	ALDP-120-9A6C	ALDP-121-044B	ALDP-122-F09A	ALDP-123-2715	ALDP-124-3625	ALDP-126-6662	ALDP-127-88CC	1A60
1A70	ALDP-128-88C6	ALDP-129-5403	ALDP-130-88C6	ALDP-131-440D	ALDP-132-88C6	ALDP-133-5413	ALDP-134-88C6	ALDP-135-441D	1A70
1A80	ALDP-136-88BC	ALDP-139-C310	ALDP-141-F00C	ALDP-142-88CC	ALDP-143-88CC	ALDP-144-17AD	ALDP-145-2243	ALDP-146-2515	1A80
1A90	ALDP-150-F73D	ALDP-154-F61F	ALDP-158-00C9	ALDP-160-F49E	ALDP-161-3713	ALDP-162-9A1C	ALDP-165-B3B2	ALDP-167-3685	1A90
1AA0	ALDP-168-E640	ALDP-171-C31C	ALDP-173-88C6	ALDP-174-54F3	ALDP-175-88C6	ALDP-176-44FD	ALDP-177-172D	ALDP-178-839C	1AA0
1AB0	ALDP-179-E711	ALDP-180-88CC	ALDP-181-25FF	ALDP-182-C490	ALDP-183-88BC	ALDP-184-9A8E	ALDP-202-0049	ALDP-204-9A96	1AB0
1AC0	ALDP-206-839C	ALDP-207-1645	ALDP-208-C74A	ALDP-209-1785	ALDP-210-5FE9	ALDP-211-5E35	ALDP-212-23FD	ALDP-213-0361	1AC0
1AD0	ALDP-214-F4D6	ALDP-215-2378	ALDP-216-23CD	ALDP-217-88CE	ALDP-218-D761	ALDP-219-5E3D	ALDP-220-176D	ALDP-221-9ACC	1AD0
1AE0	ALDP-222-174D	ALDP-223-E749	ALDP-224-22FF	ALDP-225-C4B2	ALDP-226-3743	ALDP-232-88CE	ALDP-233-1743	ALDP-234-C375	1AE0
1AF0	ALDP-235-88BC	ALDP-236-88D2	ALDP-237-2243	ALDP-238-88C4	ALDP-239-9AB4	CMLT-104-6FFB	CMLT-105-6EED	CMLT-106-677D	1AF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	1A-- 1B--
1B00	CMLT-107-666D	CMLT-108-6BBD	CMLT-109-6AAD	CMLT-110-699D	CMLT-111-688D	CMLT-112-128E	ECOL-041-75C0	ECOL-042-2DAB	1B00
1B10	ECOL-043-7BC0	ECOL-044-2DAB	ECOL-045-7FC0	ECOL-046-1C1B	ECOL-047-F871	ECOL-048-5959	ECOL-049-5AB9	ECOL-050-5EF9	1B10
1B20	ECOL-051-2DAB	ECOL-052-9B0C	ECOL-022-5738	ECOL-023-E72A	ECOL-024-6541	ECOL-025-F72E	ECOL-026-6B41	ECOL-027-C332	1B20
1B30	ECOL-028-6F41	ECOL-029-D336	ECOL-030-6941	ECOL-031-E33A	ECOL-032-6A41	ECOL-033-F33E	ECOL-034-6E41	ECOL-035-5224	1B30
1B40	ECOL-036-282B	ECOL-037-311B	ECOL-038-C48D	ECOL-039-5879	ECOL-040-B309	ECOL-003-2485	ECOL-004-2AF7	ECOL-005-5AB9	1B40
1B50	ECOL-006-4EA6	ECOL-007-5B99	ECOL-008-5B59	ECOL-009-01E3	ECOL-010-F0E1	ECOL-011-211B	ECOL-012-D861	ECOL-013-311B	1B50
1B60	ECOL-014-E824	ECOL-015-261B	ECOL-016-F4A4	ECOL-017-2DA1	ECOL-018-5EC0	ECOL-019-3E13	ECOL-020-7EC0	ECOL-021-9B7C	1B60
1B70	ECOL-053-2DC0	ECOL-054-2DFB	ECOL-055-5119	ECOL-056-C4FD	ECOL-057-08AD	ECOL-058-E0CA	ECOL-059-1C2B	ECOL-060-E814	1B70
1B80	ECOL-061-5629	ECOL-062-71A2	ECOL-063-5C99	ECOL-064-8F64	ECOL-065-8A68	ECOL-066-2E07	ECOL-067-C536	ECOL-068-F837	1B80
1B90	ECOL-069-52D9	ECOL-070-95CA	ECOL-095-5089	ECOL-096-51A2	ECOL-097-52E2	ECOL-098-D81E	ECOL-099-5224	ECOL-100-01E3	1B90
1BA0	ECOL-101-FCAB	ECOL-102-21FF	ECOL-103-D829	ECOL-104-212B	ECOL-105-1C4B	ECOL-106-58D5	ECOL-107-2DCB	ECOL-108-2D3D	1BA0
1BB0	ECOL-109-5889	ECOL-110-F0BE	ECOL-111-9B4A	ECOL-071-3485	ECOL-082-5FA0	ECOL-083-7CA8	ECOL-085-9F01	ECOL-112-2D3D	1BB0
1BC0	ECOL-113-2D2B	ECOL-114-5FC0	ECOL-115-3D9B	ECOL-116-5BC0	ECOL-117-3D9B	ECOL-118-55C0	ECOL-119-1C1B	ECOL-120-F852	1BC0
1BD0	ECOL-121-9B46	ECOL-122-5FE9	ECOL-123-5BA9	ECOL-124-5599	ECOL-125-3D9B	ECOL-126-9BC2	CCOM-071-2B73	CCOM-072-89FC	1BD0
1BE0	CSFT-046-9D3A	CSFT-047-2D2B	CSFT-048-9BE8	CSFT-049-1D23	CSFT-050-8420	CSFT-051-B340	CSFT-052-9D3A	CFLS-014-1E85	1BE0
1BF0	CFLS-021-1E8D	CFLS-022-2C25	CFLS-023-CE78	CFLS-024-1C7D	CFLS-025-B170	CFLS-026-D5FF	CFLS-027-2C07	CFLS-028-B1A6	1BF0

	0	2	4	6	8	A	C	E	1C--
1C00	ECOL-127-C822	ECOL-128-980C	ECOL-129-2445	ECOL-130-9C1E	ECOL-131-2425	ECOL-132-9C1E	ECOL-133-2415	ECOL-134-9C1E	1C00
1C10	ECOL-135-2483	ECOL-136-9C1E	ECOL-137-2443	ECOL-138-9C1E	ECOL-139-2423	ECOL-140-9C1E	ECOL-141-2413	ECOL-142-C824	1C10
1C20	ECOL-143-9B56	ECOL-144-9C9C	ECOL-145-9CD2	DCLT-008-AECA	DCLT-003-C62F	DCLT-004-2007	DCLT-005-A262	DCLT-009-24F6	1C20
1C30	DCLT-010-ECA6	DCLT-011-F026	DCLT-012-2400	DCLT-013-5202	DCLT-014-6311	DCLT-015-C4A6	DCLT-016-24F6	DCLT-017-C142	1C30
1C40	DCLT-018-3485	DCLT-019-70F2	DCLT-020-5419	DCLT-021-5209	DCLT-022-4452	DCLT-023-7432	DCLT-024-0D04	DCLT-025-70E2	1C40
1C50	DCLT-026-0D40	DCLT-027-021E	DCLT-028-2400	DCLT-029-4652	DCLT-031-2B87	DCLT-032-52AC	DCLT-033-526F	DCLT-034-566B	1C50
1C60	DCLT-036-5212	DCLT-037-50F2	DCLT-038-5EE2	DCLT-039-1F13	DCLT-040-A024	DCLT-041-8C7A	DCLT-042-5A22	DCLT-043-2A13	1C60
1C70	DCLT-044-2400	DCLT-045-A43E	EPXF-049-2D1D	EPXF-050-2DEB	EPXF-051-5BC0	EPXF-052-6B41	EPXF-053-7BC0	EPXF-054-58D5	1C70
1C80	EPXF-055-C504	EPXF-056-6141	EPXF-057-D508	EPXF-058-6741	EPXF-059-E50C	EPXF-060-6941	EPXF-061-F510	EPXF-062-6F41	1C80
1C90	EPXF-063-C114	EPXF-064-6E41	EPXF-065-282B	EPXF-066-20FF	EPXF-067-C49D	EPXF-068-9CF6	EPXF-069-71C0	EPXF-070-2DAB	1C90
1CA0	EPXF-071-77C0	EPXF-072-2DAB	EPXF-073-79C0	EPXF-074-2D5D	EPXF-075-7FC0	EPXF-076-2DAB	EPXF-077-5EF9	EPXF-078-7FC0	1CA0
1CB0	EPXF-079-58D5	EPXF-080-5019	EPXF-081-C4C6	EPXF-082-9B80	EPXF-020-5B38	EPXF-021-55A0	EPXF-022-157B	EPXF-023-9103	1CB0
1CC0	EPXF-003-5109	EPXF-007-2A65	EPXF-009-CD5B	EPXF-010-21F7	EPXF-011-5179	EPXF-012-51F9	EPXF-013-51E9	EPXF-014-5199	1CC0
1CD0	EPXF-015-2485	EPXF-016-261B	EPXF-017-F4B8	EPXF-018-5019	EPXF-019-9B66	EPXF-024-58D5	EPXF-025-5889	EPXF-026-F0C7	1CD0
1CE0	EPXF-028-51C0	EPXF-029-2DAB	EPXF-030-57C0	EPXF-031-2DAB	EPXF-032-59C0	EPXF-033-2D5D	EPXF-034-5FC0	EPXF-035-2DAB	1CE0
1CF0	EPXF-036-55C0	EPXF-037-55E9	EPXF-038-58D5	EPXF-039-5859	EPXF-040-B109	CSFT-014-10CE	CSFT-015-7CC2	CSFT-016-51D9	1CF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	1C-- 1D--
1D00	CSFT-017-55BB	CSFT-018-5EA8	CSFT-019-56AA	CSFT-020-D014	CSFT-021-F50E	CSFT-022-5CC2	CSFT-023-AE94	CSFT-024-3B15	1D00
1D10	CSFT-025-52A8	CSFT-026-50AA	CSFT-027-E01C	CSFT-028-CE1C	CSFT-029-2AF7	CSFT-030-2040	CSFT-031-5DC9	CSFT-032-6CC3	1D10
1D20	CSFT-033-1C85	CSFT-034-E0AB	CSFT-035-B29A	CSFT-036-2CFD	CSFT-037-E0A4	CSFT-038-F035	CSFT-039-D930	CSFT-040-B340	1D20
1D30	CSFT-041-1DC3	CSFT-042-8977	CSFT-054-1D83	CSFT-055-F0BB	CSFT-056-8420	CSFT-057-2A07	CSFT-058-5CC2	CSFT-059-D046	1D30
1D40	CSFT-060-72A8	CSFT-061-70AA	CSFT-062-1B15	CSFT-063-7EA8	CSFT-064-76AA	CSFT-065-E04F	CSFT-066-AA76	CSFT-067-F05E	1D40
1D50	CSFT-068-C5DE	CSFT-069-56A0	CSFT-070-1685	CSFT-071-D1DA	CSFT-072-3685	CSFT-073-76A0	CSFT-074-AC40	CSFT-075-CE64	1D50
1D60	CSFT-076-A820	CSFT-085-A824	CSFT-077-0008	CSFT-078-4EE6	CSFT-079-4666	CSFT-080-D070	CSFT-081-4226	CSFT-082-4006	1D60
1D70	CSFT-083-D5E3	CSFT-084-AA74	BWRP-124-E1F9	BWRP-125-8174	BWRP-126-2A07	BWRP-127-2B43	BWRP-128-2F07	BWRP-129-89FE	1D70
1D80	ERCX-049-B3E8	ERCX-053-998C	ERCX-002-5E88	ERCX-003-6FF3	ERCX-004-6FF3	ERCX-005-5FB3	ERCX-006-6BE5	ERCX-007-1B73	1D80
1D90	ERCX-008-5FF9	ERCX-009-E096	ERCX-010-9A71	ERCX-011-0E83	ERCX-012-F0A4	ERCX-013-3B43	ERCX-014-CF21	ERCX-015-9F71	1D90
1DA0	ERCX-016-0F85	ERCX-017-E0CD	ERCX-018-98EE	ERCX-031-3C13	ERCX-032-5629	ERCX-033-2607	ERCX-034-54E9	ERCX-035-5460	1DA0
1DB0	ERCX-036-C536	ERCX-037-F83D	ERCX-038-9B90	ERCX-039-221B	ERCX-040-F4BD	ERCX-041-3C13	ERCX-042-71A2	ERCX-044-F701	1DB0
1DC0	ERCX-045-CBC4	ERCX-046-3D83	ERCX-047-0D80	ERCX-048-9212	ERCX-019-3B23	ERCX-020-3B13	ERCX-021-5FA0	ERCX-022-7F38	1DC0
1DD0	ERCX-023-21FF	ERCX-024-C4A9	ERCX-025-261B	ERCX-026-F4A7	ERCX-027-E404	ERCX-028-5F88	ERCX-029-7F38	ERCX-030-9DD0	1DD0
1DE0	CFHA-044-6DDD	CFHA-045-655B	CFHA-046-F46F	CFHA-047-611D	CFHA-048-600D	CFHA-049-633D	CFHA-050-622D	CFHA-051-699D	1DE0
1DF0	CFHA-052-688D	CFHA-053-6FFD	CFHA-054-C1FB	CFHA-055-6AAD	CFHA-056-128E	CFHA-057-6EED	CFHA-058-128E	CFMD-009-3082	1DF0

	0	2	4	6	8	A	C	E	1E--
1E00	CFMD-010-B170	CFMD-011-D595	CFMD-012-D18D	CFMD-013-F00C	CFMD-014-27F3	CFMD-015-B1A8	CFMD-016-F011	CFMD-017-1415	1E00
1E10	CFMD-019-5CC2	CFMD-020-B1A0	CFMD-021-D19C	CFMD-022-5E69	CFMD-023-3E85	CFMD-024-9E22	CFMD-025-5E79	CFMD-026-1E85	1E10
1E20	CFMD-027-2E4D	CFMD-028-0FF3	CFMD-029-C4AC	CFMD-030-82F8	CFMD-031-2EFF	CFMD-032-9E22	CFMD-033-D1BF	CFMD-034-3B35	1E20
1E30	CFMD-035-5ED9	CFMD-036-2E07	CFMD-037-896A	CFMD-038-5AE2	CFMD-039-B42E	CFMD-040-2040	CFMD-041-9E00	CFMD-042-5EC9	1E30
1E40	CFMD-043-2E07	CFMD-044-6761	CFMD-045-17F3	CFMD-046-7CDD	CFMD-047-57D9	CFMD-048-F5CF	CFMD-049-3D13	CFMD-050-7CA2	1E40
1E50	CFMD-051-896A	CFMD-052-3B35	CFMD-053-F058	CFMD-054-9F82	CFMD-055-10CE	CFMD-056-2080	CFMD-057-F46E	CFMD-058-42E6	1E50
1E60	CFMD-059-4C86	CFMD-060-B42E	CFMD-061-2B07	CFMD-062-46E6	CFMD-063-4E86	CFMD-064-48A6	CFMD-065-8992	CFMD-066-4C26	1E60
1E70	CFMD-067-4606	CFMD-068-B42E	CFMD-069-4A86	CFMD-070-48E6	CFMD-071-4E06	CFMD-072-4066	CFMD-073-4626	CFMD-074-42C6	1E70
1E80	CFMD-075-8992	DCLC-165-95DA	DCLC-124-A154	DCLC-085-E605	DCLC-086-FD8F	DCLC-087-262B	DCLC-088-FD86	DCLC-089-5F4F	1E80
1E90	DCLC-090-FC85	DCLC-091-E123	DCLC-092-F123	DCLC-093-2B88	DCLC-094-E605	DCLC-095-C9A0	DCLC-096-262B	DCLC-097-C999	1E90
1EA0	DCLC-098-8E4C	DCLC-099-C4B6	DCLC-100-F233	DCLC-101-2D04	DCLC-102-3685	DCLC-103-7632	DCLC-104-1685	DCLC-105-D134	1EA0
1EB0	DCLC-106-2D02	DCLC-107-1543	DCLC-108-8442	DCLC-126-C04E	DCLC-127-D03C	DCLC-128-0D40	DCLC-129-F44F	DCLC-130-2D04	1EB0
1EC0	DCLC-131-D148	DCLC-132-F247	DCLC-133-2D02	DCLC-134-1543	DCLC-135-2285	DCLC-136-2307	DCLC-137-7232	DCLC-138-2307	1EC0
1ED0	DCLC-139-C55F	DCLC-140-D560	DCLC-141-04C7	DCLC-142-C4E0	DCLC-143-C461	DCLC-144-D065	DCLC-145-C07B	DCLC-146-15C5	1ED0
1EE0	DCLC-147-0D10	DCLC-148-A782	DCLC-154-2D10	DCLC-155-2B88	DCLC-156-E605	DCLC-157-C9F0	DCLC-158-262B	DCLC-159-C9E9	1EE0
1EF0	DCLC-160-2645	DCLC-161-3613	DCLC-162-5212	DCLC-163-D402	DCLC-164-95D6	DCLC-149-2D10	DCLC-150-2B48	DCLC-151-2405	1EF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	1E-- 1F--
1F00	DCLC-152-1573	DCLC-153-AF72	ERRQ-054-8CA4	ERRQ-064-2406	ERRQ-065-0B30	ERRQ-066-9816	ERRQ-067-8A68	ERRQ-068-999E	1F00
1F10	ERRQ-013-C468	ERRQ-014-CFB5	ERRQ-015-8CD4	ERRQ-055-E261	ERRQ-056-C3C5	ERRQ-058-C422	ERRQ-059-D15F	ERRQ-060-CA23	1F10
1F20	ERRQ-061-DA5F	ERRQ-062-AB20	ERRQ-004-8F70	ERRQ-005-3623	ERRQ-006-CEAD	ERRQ-007-9936	ERRQ-008-F710	ERRQ-009-C439	1F20
1F30	ERRQ-010-DD94	ERRQ-011-D995	ERRQ-012-810C	ERRQ-030-8F6C	ERRQ-031-2466	ERRQ-032-E607	ERRQ-033-2406	ERRQ-034-D996	1F30
1F40	ERRQ-035-0B10	ERRQ-036-3485	ERRQ-037-8FCE	ERRQ-038-D54A	ERRQ-039-8D64	ERRQ-040-3E43	ERRQ-041-E261	ERRQ-042-F75E	1F40
1F50	ERRQ-043-C3DE	ERRQ-044-D45D	ERRQ-045-EA5D	ERRQ-046-4452	ERRQ-047-B406	ERRQ-048-A7F2	ERRQ-049-A7D8	ERRQ-063-8D68	1F50
1F60	ERRQ-050-2A07	ERRQ-051-8FCE	ERRQ-052-D104	ERRQ-053-8CDE	ERRQ-017-8F64	ERRQ-018-DB96	ERRQ-019-0F10	ERRQ-020-C442	1F60
1F70	ERRQ-021-FFC4	ERRQ-022-9816	ERRQ-023-2D15	ERRQ-024-5CCD	ERRQ-025-B3E8	ERRQ-026-7CA0	ERRQ-027-1545	ERRQ-028-3E23	1F70
1F80	ERRQ-029-9F44	CFDV-006-3082	CFDV-007-B04A	CFDV-008-5AE2	CFDV-009-B42E	CFDV-010-46A6	CFDV-011-3B35	CFDV-012-3713	1F80
1F90	CFDV-013-76F2	CFDV-014-F5A2	CFDV-015-2C13	CFDV-016-A656	CFDV-017-5CA2	CFDV-018-2C18	CFDV-019-C4A0	CFDV-020-1D13	1F90
1FA0	CFDV-021-7CA2	CFDV-022-9DE0	CFDV-023-3002	CFDV-024-0040	CFDV-025-2D13	CFDV-026-B04A	CFDV-027-F5EE	CFDV-028-9DE0	1FA0
1FB0	CFDV-029-3002	CFDV-030-D1BB	CFDV-031-CD2A	CFDV-032-2040	CFDV-033-9FAA	CFDV-037-56F2	CFDV-038-7D68	CFDV-039-76F2	1FB0
1FC0	CFDV-040-F448	CFDV-041-07CB	CFDV-042-F0A6	CFDV-043-9FCC	CFDV-044-5779	CFDV-045-F0A6	CFDV-046-5AE2	CFDV-047-B42E	1FC0
1FD0	CFDV-048-B170	CFDV-049-D5D7	CFDV-050-B1A0	CFDV-051-56A2	CFDV-052-5659	CFDV-053-5CC2	CFDV-054-1585	CFDV-055-C762	1FD0
1FE0	CFDV-056-3585	CFDV-057-75A0	CFDV-058-F375	CFDV-059-C66B	CFDV-060-0080	CFDV-061-27C3	CFDV-062-B1A8	CFDV-034-9DE0	1FE0
1FF0	CFDV-035-1002	CFDV-036-9FB2	CFDV-063-27D3	CFDV-064-55E9	CFDV-065-B19E	5007	BDIA-087-2810	BDIA-088-D57F	1FF0

	0	2	4	6	8	A	C	E	20--
2000	BDIA-091-2810	BDIA-092-25F5	BDIA-096-E504	BDIA-097-2810	BDIA-101-F508	BDIA-102-2810	BDIA-103-A040	CCOM-110-2607	2000
2010	CCOM-114-577B	CCOM-115-3763	CCOM-116-5B60	CCOM-117-5B6B	CCOM-118-2707	CCOM-120-F0A5	CCOM-121-4E52	CCOM-123-4252	2010
2020	CCOM-125-4626	CCOM-126-4026	CCOM-127-2B45	CCOM-129-42AF	CCOM-131-76B8	CCOM-132-72B8	CCOM-133-6AA4	CCOM-134-70B0	2020
2030	CCOM-135-2B47	CCOM-137-42AF	CCOM-139-7EB0	CCOM-141-5282	CCOM-142-423F	CCOM-145-C58E	CCOM-146-128E	CCOM-147-AED0	2030
2040	BDIA-106-2810	BDIA-111-C55F	FILE-344-2C07	FILE-345-4BCF	FILE-346-4ECF	FILE-347-2D15	FILE-348-4BDF	FILE-349-2D45	2040
2050	FILE-350-4EDF	FILE-351-2D07	FILE-352-4BDF	FILE-353-4EDF	FILE-354-4FDF	FILE-355-49DF	FILE-356-128E	BDIA-114-2810	2050
2060	BDIA-119-D57F	CMLT-006-AE94	CMLT-002-F563	CMLT-003-3515	CMLT-004-B3F6	CMLT-005-897A	CMLT-007-E1F1	CMLT-008-3485	2060
2070	CMLT-009-F563	CMLT-010-3515	CMLT-011-8270	CMLT-012-897A	CBIN-072-8270	CBIN-073-8E7C	5007	BDIA-125-2800	2070
2080	BDIA-131-3E09	BDIA-132-C483	BDIA-133-F085	BDIA-134-E087	BDIA-135-F489	BDIA-136-6EE5	BDIA-137-A0DA	BDIA-138-2E0D	2080
2090	BDIA-142-A0DA	BDIA-143-7EE1	BDIA-147-F494	BDIA-148-7EE1	BDIA-149-6EE3	BDIA-150-2E5F	BDIA-151-6EE3	BDIA-152-6EE9	2090
20A0	BDIA-153-0E5F	BDIA-154-C4A2	BDIA-155-2E87	BDIA-156-3E27	BDIA-157-2EEF	BDIA-158-2E77	BDIA-159-1E87	BDIA-160-1EB7	20A0
20B0	BDIA-161-2EF7	BDIA-162-0E11	BDIA-163-F0B4	BDIA-164-E0B6	BDIA-165-1EFB	BDIA-166-A0E2	BDIA-167-1EF3	BDIA-171-A0E2	20B0
20C0	BDIA-172-3E15	BDIA-176-A0E2	BDIA-177-3E0D	BDIA-181-2E1D	BDIA-182-2FF5	BDIA-183-A0DA	BDIA-184-0FFD	BDIA-188-C4CE	20C0
20D0	BDIA-189-1FF5	BDIA-190-C4D2	BDIA-191-2EF5	BDIA-192-A0E2	BDIA-193-A0E8	BDIA-195-0EFF	BDIA-196-C4E1	BDIA-197-2810	20D0
20E0	BDIA-198-128E	BDIA-201-0EFD	BDIA-202-C4DE	BDIA-203-128E	BDIA-207-24F6	BDIA-208-578F	BDIA-209-1887	BDIA-210-087F	20E0
20F0	BDIA-211-E0F0	BDIA-212-F0F2	BDIA-213-2400	BDIA-214-574F	BDIA-215-0487	BDIA-216-E0FA	BDIA-217-F0FC	BDIA-221-2EF7	20F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	20-- 21--
2100	BDIA-222-2FF7	BDIA-223-6E02	BDIA-224-2E05	BDIA-225-2F05	BDIA-226-4E02	BDIA-227-0FFF	BDIA-228-E08C	BDIA-229-F08E	2100
2110	BDIA-230-1EFF	BDIA-231-C492	BDIA-232-2E33	BDIA-233-2F87	BDIA-234-6FE8	BDIA-235-6FEA	BDIA-236-44E8	BDIA-237-2F13	2110
2120	BDIA-238-048F	BDIA-239-E0A2	BDIA-240-F0A4	BDIA-241-257F	BDIA-242-C4A8	BDIA-243-2E05	BDIA-244-55EA	BDIA-245-C4AE	2120
2130	BDIA-246-2E45	BDIA-247-3FE9	BDIA-248-25F7	BDIA-249-75E0	BDIA-251-2505	BDIA-252-55EA	BDIA-254-05FF	BDIA-255-C4BE	2130
2140	BDIA-256-FB34	BDIA-257-44E6	BDIA-258-044D	BDIA-259-C4C6	BDIA-260-75F1	BDIA-261-C4CA	BDIA-262-0C60	BDIA-263-88EC	2140
2150	CFLS-007-1E85	CFLS-008-9BF2	DCLR-155-562F	DCLR-156-221E	DCLR-157-5C3F	DCLR-159-7222	DCLR-163-5D2F	DCLR-164-5E3F	2150
2160	DCLR-165-2B10	DCLR-166-2407	DCLR-167-2585	DCLR-168-3573	DCLR-169-765A	DCLR-170-725A	DCLR-171-5350	DCLR-172-231B	2160
2170	DCLR-173-33C5	DCLR-175-5622	DCLR-179-5729	DCLR-180-725A	DCLR-181-577F	DCLR-182-7650	DCLR-183-3573	DCLR-184-5650	2170
2180	DCLR-188-E690	DCLR-189-2B00	DCLR-190-A194	DCLR-229-5349	DCLR-230-3613	DCLR-231-1D00	DCLR-232-B026	DCLR-152-A7D6	2180
2190	DCLR-191-24F6	DCLR-192-2B80	DCLR-193-2507	DCLR-194-021E	DCLR-195-2323	DCLR-196-0D02	DCLR-197-C38F	DCLR-198-2B80	2190
21A0	DCLR-199-C9B8	DCLR-200-2B84	DCLR-201-2207	DCLR-202-C9B8	DCLR-203-221B	DCLR-204-F4A6	DCLR-205-2D10	DCLR-206-3B80	21A0
21B0	DCLR-207-221B	DCLR-208-C9B8	DCLR-209-F4B0	DCLR-210-0D10	DCLR-212-C687	DCLR-213-E687	DCLR-217-F207	DCLR-218-221E	21B0
21C0	DCLR-219-7222	DCLR-220-D8C6	DCLR-221-3413	DCLR-222-2B00	DCLR-223-2400	DCLR-224-5E22	DCLR-225-021E	DCLR-226-1000	21C0
21D0	DCLR-227-B046	DYPE-090-5AFF	DYPE-091-1FC5	DYPE-092-DAF1	DYPE-093-CADC	DYPE-094-2D15	DYPE-095-C06D	DYPE-096-4806	21D0
21E0	DYPE-097-957A	DYPE-098-4086	DYPE-099-42CF	DYPE-100-F56B	DYPE-101-7F38	DYPE-102-87D2	DYPE-112-87DE	DYPE-111-B36E	21E0
21F0	DYPE-103-0F5B	DYPE-104-C4FD	DYPE-105-0FAB	DYPE-106-C4DA	DYPE-107-2E13	DYPE-108-1545	DYPE-109-C06F	DYPE-110-87E0	21F0

	0	2	4	6	8	A	C	E	22--
2200	CMLT-158-3D25	CMLT-159-A752	CMLT-160-1D25	CMLT-161-A754	CMLT-162-89C4	DMCS-115-1028	DMCS-115-303C	DMCS-115-5078	2200
2210	DMCS-115-4C5D	DMCS-115-6C4D	DMCS-115-7C7D	DMCS-115-7B7E	DCLT-047-3000	DCLT-049-2480	DCLT-050-5432	DCLT-051-C425	2210
2220	DCLT-052-E9A8	DCLT-053-A6FC	DCLT-055-D9A3	DCLT-056-C9A3	DCLT-060-5A02	DCLT-061-2A15	DCLT-062-7A22	DCLT-063-2A13	2220
2230	DCLT-064-C464	DCLT-065-2A07	DCLT-066-2413	DCLT-067-24F6	DCLT-068-221E	DCLT-069-F074	DCLT-070-1413	DCLT-071-9C3E	2230
2240	DCLH-015-8DF6	DCLH-016-2C25	DCLH-017-24F6	DCLH-018-2210	DCLH-019-F04E	DCLH-020-0210	DCLH-021-810C	DCLH-022-1D00	2240
2250	DCLH-023-2884	DCLH-024-0210	DCLH-025-E55D	DCLH-026-4006	DCLH-027-C4DD	DCLH-028-3445	DCLH-029-B02E	DCLH-004-C643	2250
2260	DCLH-005-2043	DCLH-006-7022	DCLH-007-24F6	DCLH-008-2B80	DCLH-009-5622	DCLH-C10-5659	DCLH-011-2645	DCLH-012-2407	2260
2270	DCLH-013-D140	DCLH-014-8DFA	DCLT-072-1583	DCLT-073-FCFA	DCLT-074-3483	DCLT-075-0D40	DCLT-076-2470	DCLT-077-74A2	2270
2280	DCLT-078-7CF2	DCLT-079-2400	DCLT-080-5BA2	DCLT-081-3B85	DCLT-082-7AE2	DCLT-083-9C52	BSYS-002-4E42	BSYS-004-2460	2280
2290	BSYS-005-2607	BSYS-023-24F6	BSYS-024-4052	BSYS-025-4206	BSYS-026-4406	BSYS-027-4606	BSYS-028-4A06	BSYS-032-2D08	2290
22A0	BSYS-042-2490	BSYS-043-8F90	BSYS-044-A044	BSYS-045-49CF	BSYS-046-E5A3	BSYS-047-2E08	BSYS-048-1E00	BSYS-060-2482	22A0
22B0	BSYS-061-2705	BSYS-063-2675	BSYS-067-5760	BSYS-068-497F	BSYS-069-2F00	BSYS-070-2400	BSYS-071-6EE4	BSYS-072-7AEA	22B0
22C0	BSYS-074-7AE0	BSYS-075-2F1D	BSYS-076-F4C0	BSYS-077-2F91	BSYS-078-7AEA	BSYS-093-2E15	BSYS-094-7AEA	BSYS-095-7AEA	22C0
22D0	BSYS-096-7AE0	BSYS-115-7A32	BSYS-117-7A52	BSYS-118-7A62	BSYS-173-40A6	BSYS-174-3212	BSYS-175-1210	BSYS-176-20A0	22D0
22E0	BSYS-177-C3E4	BSYS-178-B390	BSYS-179-E8EE	BSYS-180-062C	BSYS-181-1212	BSYS-182-4052	BSYS-183-8A26	BSYS-184-162C	22E0
22F0	BSYS-185-1212	BSYS-186-5C82	BSYS-187-44CF	BSYS-188-42DF	BSYS-189-4852	BSYS-190-5CC2	BSYS-191-2F07	BSYS-192-8A40	22F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	22-- 23--
2300	CFMU-062-C6C5	CFMU-063-0080	CFMU-064-27C3	CFMU-065-B1A6	CFMU-024-5CA2	CFMU-025-C91B	CFMU-026-3D83	CFMU-027-7CA2	2300
2310	CFMU-028-5CE2	CFMU-029-52C8	CFMU-030-5EC0	CFMU-031-2C13	CFMU-032-89D6	CFMU-033-54B2	CFMU-034-5AF9	CFMU-035-5B89	2310
2320	CFMU-036-5699	CFMU-037-5359	CFMU-038-5039	CFMU-039-5109	CFMU-040-5219	CFMU-041-5729	CFMU-042-46C6	CFMU-043-5CC2	2320
2330	CFMU-044-0FF3	CFMU-045-C4BC	CFMU-046-B2F8	CFMU-047-26FF	CFMU-048-F4BD	CFMU-049-3713	CFMU-050-5AE2	CFMU-051-B170	2330
2340	CFMU-052-05C5	CFMU-053-B1A0	CFMU-054-56E9	CFMU-055-1E85	CFMU-056-C74C	CFMU-057-3E85	CFMU-058-2080	CFMU-059-F300	2340
2350	CFMU-060-27D3	CFMU-061-B19E	CFMU-009-5CB2	CFMU-010-FC08	CFMU-011-44C6	CFMU-012-1415	CFMU-013-5CA2	CFMU-014-5079	2350
2360	CFMU-015-5FC9	CFMU-016-57F9	CFMU-017-5189	CFMU-018-5299	CFMU-019-5329	CFMU-020-5E39	CFMU-021-2107	CFMU-022-2507	2360
2370	CFMU-023-A32C	CMLT-163-3002	CMLT-164-A752	CMLT-165-1002	CMLT-166-3D25	CMLT-167-A754	CMLT-168-89C4	DMCS-003-1E7D	2370
2380	DMCS-004-E09C	DMCS-005-E81C	DMCS-006-EB09	DMCS-007-F21D	DMCS-009-2C75	DMCS-013-2D05	DMCS-014-53C0	DMCS-015-1385	2380
2390	DMCS-016-CA15	DMCS-017-3385	DMCS-018-73C0	DMCS-019-8C58	DMCS-020-EE4A	DMCS-021-EB1F	DMCS-022-8C5C	DMCS-026-74E2	2390
23A0	DMCS-027-76F2	DMCS-028-2F10	DMCS-032-2713	DMCS-033-72A2	DMCS-034-2427	DMCS-035-25A3	DMCS-036-4148	DMCS-037-51C9	23A0
23B0	DMCS-038-4826	DMCS-039-691B	DMCS-040-68AD	DMCS-041-5B38	DMCS-042-5D98	DMCS-043-2C1B	DMCS-044-6BD1	DMCS-045-C4D1	23B0
23C0	DMCS-046-52A2	DMCS-047-6773	DMCS-048-D72C	DMCS-049-21F5	DMCS-050-A3D4	DMCS-023-F21D	DMCS-024-3F23	DMCS-025-8C58	23C0
23D0	DMCS-051-0CFD	DMCS-052-C4B6	DMCS-053-497F	DMCS-055-2C75	DMCS-059-2D05	DMCS-060-55C0	DMCS-061-C560	DMCS-062-3785	23D0
23E0	DMCS-063-2505	DMCS-064-52A2	DMCS-068-2BF7	DMCS-069-7BC8	DMCS-070-F866	DMCS-071-2B05	DMCS-072-B248	CBIN-018-AE06	23E0
23F0	CBIN-019-AD54	DCHN-041-56D2	DCHN-053-1773	DCHN-059-57B5	DCHN-060-1B83	DCHN-061-3BB5	DCHN-063-5BA0	DCHN-073-5468	23F0

	0	2	4	6	8	A	C	E	24--
2400	DCHN-074-C251	DCHN-075-B44A	DCHN-076-4052	DCHN-077-4206	DCHN-078-D146	DCHN-079-C110	DCHN-080-3485	DCHN-081-1583	2400
2410	DCHN-082-556A	DCHN-083-D119	DCHN-084-C118	DCHN-085-3085	DCHN-086-5666	DCHN-087-7068	DCHN-088-7160	DCHN-089-176B	2410
2420	DCHN-090-5C68	DCHN-091-5260	DCHN-092-021E	DCHN-093-0080	DCHN-094-55E9	DCHN-095-5F59	DCHN-096-54F9	DCHN-097-5B49	2420
2430	DCHN-098-D235	DCHN-099-3000	DCHN-100-2400	DCHN-101-C1C3	DCHN-102-A00E	DCHN-103-54B9	DCHN-104-55F9	DCHN-105-2135	2430
2440	DCHN-106-8A00	DCHN-120-A01E	DCHN-121-A43A	DCHN-107-5560	DCHN-108-716A	DCHN-109-7060	DCHN-110-2080	DCHN-111-A428	2440
2450	DCHN-112-5666	DCHN-113-0010	DCHN-114-1583	DCHN-115-1413	DCHN-116-7468	DCHN-117-2485	DCHN-118-2505	DCHN-119-A41E	2450
2460	DYPE-227-8A9E	DYPE-232-D56A	DYPE-233-3E83	DYPE-234-6664	DYPE-235-7E6A	DYPE-236-8AC0	DYPE-221-8F7A	DYPE-222-2404	2460
2470	DYPE-223-C463	DYPE-224-D179	DYPE-225-EEE0	DYPE-226-ADF6	DYPE-228-1F00	DYPE-229-2F08	DYPE-230-3483	DYPE-231-8AC2	2470
2480	ETRP-079-23B1	ETRP-080-2D07	ETRP-081-7D20	ETRP-082-A4D8	ETRP-023-FADA	ETRP-024-5ACF	ETRP-025-5BFF	ETRP-026-5F2C	2480
2490	ETRP-027-F245	ETRP-029-D216	ETRP-030-DFAB	ETRP-031-236D	ETRP-032-23CB	ETRP-033-5D20	ETRP-034-7F20	ETRP-035-337F	2490
24A0	ETRP-036-5F20	ETRP-037-1FFF	ETRP-038-6DF1	ETRP-039-7D20	ETRP-040-23CB	ETRP-041-223D	ETRP-045-5F20	ETRP-046-6CF1	24A0
24B0	ETRP-047-C4C3	ETRP-048-CAC2	ETRP-049-3F40	ETRP-050-2E55	ETRP-051-54FD	ETRP-052-3FF5	ETRP-053-5DE0	ETRP-054-6DC5	24B0
24C0	ETRP-055-7DE0	ETRP-056-2215	ETRP-057-241B	ETRP-058-F0D0	ETRP-059-F24C	ETRP-060-1613	ETRP-061-246B	ETRP-062-E0D3	24C0
24D0	ETRP-063-0208	ETRP-065-DF80	ETRP-069-CFD8	ETRP-070-2D10	ETRP-071-0D80	ETRP-072-2215	ETRP-073-2307	ETRP-074-5F2C	24D0
24E0	ETRP-075-3613	ETRP-076-2435	ETRP-077-A4CC	BCHK-107-26C5	BCHK-127-2705	BCHK-128-4452	BCHK-129-4248	BCHK-130-7268	24E0
24F0	BCHK-131-F06C	BCHK-175-F679	BCHK-192-26D5	BCHK-210-A4EA	BCHK-103-4A52	BCHK-104-A28C	CBIN-022-8270	CBIN-023-AC1A	24F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	24-- 25--
2500	CDMD-077-7D30	CDMD-078-7EA3	CDMD-079-C4CF	CDMD-080-B3C8	CDMD-081-0D9D	CDMD-082-E08E	CDMD-083-2040	CDMD-084-5D30	2500
2510	CDMD-085-5D78	CDMD-086-7D9F	CDMD-087-47DB	CDMD-088-7D30	CDMD-089-D186	CDMD-090-A53C	CDMD-052-B3C8	CDMD-053-C1A4	2510
2520	CDMD-054-8174	CDMD-055-9BDC	CDMD-056-F5B0	CDMD-057-1002	CDMD-058-B3CA	CDMD-059-8174	CDMD-060-2BB3	CDMD-061-89FC	2520
2530	CDMD-062-56A2	CDMD-063-5D70	CDMD-064-5DDB	CDMD-065-7D70	CDMD-066-2993	CDMD-067-5981	CDMD-068-5004	CDMD-069-1042	2530
2540	CDMD-070-56E2	CDMD-071-B3CC	CDMD-072-5D30	CDMD-073-7D8F	CDMD-074-F081	CDMD-075-7D30	CDMD-076-A53E	CDMD-091-C1A1	2540
2550	CDMD-092-5CC2	CDMD-093-2FC3	CDMD-094-C958	CDMD-095-1F6B	CDMD-096-E45D	CDMD-097-3F13	CDMD-098-5730	CDMD-099-67F5	2550
2560	CDMD-100-7730	CDMD-101-6443	CDMD-102-8DC2	DYPE-179-2791	DYPE-180-566D	DYPE-181-16E3	DYPE-182-2404	DYPE-183-5C68	2560
2570	DYPE-184-5463	DYPE-185-5268	DYPE-186-5068	DYPE-187-52E9	DYPE-188-2771	DYPE-189-128E	CLOG-114-8270	CLOG-115-8CE6	2570
2580	DYPE-151-1547	DYPE-152-8378	DYPE-153-8C70	DYPE-170-3F20	DYPE-171-E88C	DYPE-172-4FFF	DYPE-173-87DA	DYPE-163-3523	2580
2590	DYPE-164-2F08	DYPE-165-87DA	DYPE-159-2EC3	DYPE-161-D558	DYPE-162-8D62	DYPE-114-9816	DYPE-115-54F9	DYPE-116-6664	2590
25A0	DYPE-117-8380	DYPE-118-A566	DYPE-119-1485	DYPE-120-1C83	DYPE-121-837A	DYPE-122-2E07	DYPE-123-5FF9	DYPE-124-C4B2	25A0
25B0	DYPE-125-AB20	DYPE-126-1D45	DYPE-127-1F28	DYPE-128-EEBA	DYPE-129-3D45	DYPE-130-0F4B	DYPE-131-C4DD	DYPE-132-EED5	25B0
25C0	DYPE-133-0F3B	DYPE-134-C495	DYPE-135-2D07	DYPE-136-0F1B	DYPE-137-C491	DYPE-138-0F9B	DYPE-139-C48F	DYPE-140-0FAB	25C0
25D0	DYPE-141-C487	DYPE-146-3D85	DYPE-147-2E23	DYPE-148-15C5	DYPE-149-F200	DYPE-150-87F4	DYPE-174-3C83	DYPE-175-837A	25D0
25E0	DYPE-176-5DF9	DYPE-177-8C3E	FILE-272-2C15	FILE-273-2D07	FILE-274-7C62	FILE-275-2A15	FILE-276-3A25	FILE-277-2B07	25E0
25F0	FILE-278-7A52	FILE-279-2A07	FILE-280-2E23	FILE-281-49BF	FILE-282-128E	5007	DUMP-049-3BF5	DUMP-050-0B61	25F0

	0	2	4	6	8	A	C	E	26--
2600	DUMP-051-F486	DUMP-052-2B7B	DUMP-053-2BCD	DUMP-054-5DAC	DUMP-055-4FEF	DUMP-056-2D04	DUMP-057-2E7B	DUMP-058-4BDF	2600
2610	DUMP-059-2EAD	DUMP-060-F497	DUMP-061-6EF3	DUMP-062-2E2B	DUMP-063-251B	DUMP-064-55C9	DUMP-065-CC21	DUMP-066-128E	2610
2620	DUMP-067-2B45	DUMP-068-2C2B	DUMP-069-0CBD	DUMP-070-C486	DUMP-071-C031	DUMP-072-241B	DUMP-073-C030	DUMP-074-5029	2620
2630	DUMP-075-2F93	DUMP-076-3D02	DUMP-077-FEB8	DUMP-078-3F85	DUMP-079-1D00	DUMP-080-4DFF	DUMP-081-FABC	DUMP-082-DB42	2630
2640	DUMP-083-1D00	DUMP-084-0D0A	DUMP-085-51E9	DUMP-086-6E21	DUMP-087-C4D3	DUMP-088-DEBC	DUMP-090-02CD	DUMP-098-C4D3	2640
2650	DUMP-099-B1BE	DUMP-100-8174	DUMP-101-AB20	CFCO-007-F45C	CFCO-008-4953	CFCO-009-F46F	CFCO-010-4153	CFCO-011-5115	2650
2660	CFCO-012-4013	CFCO-013-5005	CFCO-014-4303	CFCO-015-5335	CFCO-016-4233	CFCO-017-5225	CFCO-018-4923	CFCO-019-5995	2660
2670	CFCO-020-4893	CFCO-021-5885	CFCO-022-4F83	CFCO-023-5FF5	CFCO-024-4AF3	CFCO-025-2CFF	CFCO-026-C4D6	CFCO-027-128E	2670
2680	ALDP-307-D648	ALDP-308-0FFD	ALDP-309-E095	ALDP-310-1FCD	ALDP-311-E08F	ALDP-312-1F4D	ALDP-313-E0B6	ALDP-314-F0B7	2680
2690	ALDP-315-2F9B	ALDP-316-E0B6	ALDP-317-5F4D	ALDP-318-A6C8	ALDP-291-5A4F	ALDP-292-3683	ALDP-293-CAAF	ALDP-294-14C5	2690
26A0	ALDP-295-DAB2	ALDP-296-04AB	ALDP-297-C4AF	ALDP-298-1623	ALDP-299-045B	ALDP-300-C4B2	ALDP-301-C632	ALDP-302-1F28	26A0
26B0	ALDP-303-9A0C	ALDP-304-957A	ALDP-305-E201	ALDP-306-9A1A	ALDP-275-3BE9	ALDP-276-56AA	ALDP-277-52AA	ALDP-278-54AA	26B0
26C0	ALDP-279-42AF	ALDP-280-C218	ALDP-281-D349	ALDP-282-4F3F	ALDP-283-2440	ALDP-284-5872	ALDP-285-2404	ALDP-286-50A0	26C0
26D0	ALDP-287-2B2D	ALDP-288-5EA0	ALDP-289-1F28	ALDP-290-128E	CICY-092-5098	CICY-093-50BB	CICY-094-C4F9	CICY-095-5EA8	26D0
26E0	CICY-096-56A0	CICY-097-617B	CICY-098-606F	CICY-099-6FAD	CICY-100-55B3	CICY-101-C4F7	CICY-102-52A8	CICY-103-56A0	26E0
26F0	CICY-104-617B	CICY-105-606D	CICY-106-6F3D	CICY-107-128E	CICY-108-2F07	CICY-109-C4E9	FINT-002-7CC2	FINT-003-5F59	26F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	26-- 27--
2700	FINT-004-D98E	FINT-005-5A32	FINT-006-2A13	FINT-007-7A32	FINT-008-9130	FINT-009-A024	FINT-010-A748	FINT-011-C996	2700
2710	FINT-012-2E07	FINT-013-9138	FINT-014-A70A	FINT-015-2B43	FINT-016-4FBF	FINT-017-5AA0	FINT-018-2A13	FINT-019-7A32	2710
2720	FINT-020-F1A8	FINT-021-0D02	FINT-022-2E25	FINT-023-A740	FINT-024-E5AF	FINT-025-A5E4	FINT-026-A740	FINT-027-8F90	2720
2730	FINT-028-4DBD	FINT-029-7A32	FINT-030-FABB	FINT-031-A044	FINT-032-A73E	FINT-033-A044	FINT-034-A5EC	FINT-035-3E43	2730
2740	FINT-036-2F07	FINT-037-2400	FINT-038-2A07	FINT-039-A01E	FINT-040-5A32	FINT-041-2A13	FINT-042-1000	FINT-043-5CC2	2740
2750	FINT-044-A43C	CMLT-113-1CFB	CMLT-114-3DE3	CMLT-115-58CA	CMLT-116-719D	CMLT-117-708D	CMLT-118-58CA	CMLT-119-779D	2750
2760	CMLT-120-768D	CMLT-121-58CA	CMLT-122-7B9D	CMLT-123-7A8D	CMLT-124-58CA	CMLT-125-759D	CMLT-126-748D	CMLT-127-1CFB	2760
2770	CMLT-128-0002	CMLT-129-F87B	CMLT-130-C5D4	CMLT-131-2002	CMLT-132-C5D5	CMLT-133-128E	CBIN-038-8270	CBIN-039-88DC	2770
2780	DCLC-208-99F6	DCLC-166-C3B2	DCLC-167-A7DC	DCLC-191-A154	DCLC-186-5429	DCLC-187-5349	DCLC-188-EC93	DCLC-189-D892	2780
2790	DCLC-190-A7A2	DCLC-192-1635	DCLC-193-7602	DCLC-194-3413	DCLC-195-2D40	DCLC-196-ECA2	DCLC-197-2B08	DCLC-198-3413	2790
27A0	DCLC-199-A7A6	DCLC-200-2B02	DCLC-201-1635	DCLC-202-E607	DCLC-203-C9AE	DCLC-204-262B	DCLC-205-C9A7	DCLC-206-E680	27A0
27B0	DCLC-207-0244	DCLC-168-F209	DCLC-169-2B88	DCLC-170-5429	DCLC-171-C13C	DCLC-172-3385	DCLC-173-7222	DCLC-174-2007	27B0
27C0	DCLC-175-7032	DCLC-176-0D04	DCLC-177-E607	DCLC-178-C9CC	DCLC-179-262B	DCLC-180-C9C5	DCLC-181-0000	DCLC-182-2B00	27C0
27D0	DCLC-183-2400	DCLC-184-5E22	DCLC-185-A030	DCLL-025-0000	DCLL-026-5007	DCLL-027-A7D8	DCLL-028-0313	DCLL-029-C4D6	27D0
27E0	DCLL-031-04C3	DCLL-032-C4D6	DCLL-033-D557	DCLL-034-0D04	DCLL-035-2407	DCLL-036-7432	DCLL-037-2B08	DCLL-038-C9EF	27E0
27F0	DCLL-039-0214	DCLL-040-2400	DCLL-041-8ADA	DCLL-042-10F5	DCLL-043-2B23	DCLL-044-70B0	DCLL-045-A2E6	5007	27F0

	0	2	4	6	8	A	C	E	28--
2800	CTRT-022-2B33	CTRT-023-2B2D	CTRT-024-7FA0	CTRT-025-2BFD	CTRT-026-70AA	CTRT-027-2707	CTRT-028-77A0	CTRT-029-25FF	2800
2810	CTRT-030-F495	CTRT-031-A824	CTRT-032-A820	CLOG-056-AB25	CLOG-057-5F10	CLOG-058-7F51	CLOG-059-C4D7	CLOG-060-F4A5	2810
2820	CLOG-061-2C55	CLOG-062-8185	CLOG-063-2C25	CLOG-064-8185	CLOG-067-6F75	CLOG-068-AB25	CLOG-069-5738	CLOG-070-5F18	2820
2830	CLOG-071-7F71	CLOG-072-C49E	CLOG-073-25FF	CLOG-074-F4AD	CLOG-075-A856	CLOG-082-B3F6	CLOG-083-6F17	CLOG-084-6E07	2830
2840	CLOG-085-6737	CLOG-086-6627	CLOG-087-76A8	CLOG-088-7EAA	CLOG-089-E5A1	CLOG-090-A856	CLOG-093-AB25	CLOG-094-5F10	2840
2850	CLOG-095-6F57	CLOG-096-7F10	CLOG-097-AB21	CLOG-098-2C07	CLOG-099-8185	CLOG-174-AB25	CLOG-175-5F10	CLOG-176-6F57	2850
2860	CLOG-177-C4D7	CLOG-178-6F51	CLOG-179-AB21	CLOG-180-2C75	CLOG-181-8185	CLOG-050-B3F6	CLOG-051-B460	CLOG-052-D5D6	2860
2870	CLOG-053-F5A5	CLOG-054-A820	CLOG-184-AB25	CLOG-185-558B	CLOG-186-2B3B	CLOG-187-5710	CLOG-188-77A0	CLOG-189-8185	2870
2880	CFCY-052-A150	CFCY-053-9BEE	CFCY-054-9BF2	CFCY-055-9BF0	CFCY-056-B17E	CFCY-057-B484	CFCY-058-B484	CFCY-059-B484	2880
2890	CFCY-060-B1A6	CFCY-061-8AF2	CFCY-062-8AF2	CFCY-063-8AF2	CFCY-064-9E00	CFCY-065-9DFE	CFCY-066-8AF2	CFCY-067-8AF2	2890
28A0	CFCY-034-A6D8	CFCY-035-2B83	CFCY-036-9814	CFCY-037-0565	CFCY-038-E0C2	CFCY-039-AB25	CFCY-040-F43F	CFCY-041-0183	28A0
28B0	CFCY-042-F0C2	CFCY-043-C038	CFCY-044-B454	CFCY-045-A8D4	CFCY-046-04F5	CFCY-047-F08A	CFCY-048-8924	CFCY-049-01C3	28B0
28C0	CFCY-050-F083	CFCY-051-AE94	CFCY-012-E1C9	CFCY-013-3485	CFCY-014-0567	CFCY-015-C4C2	CFCY-016-9814	CFCY-017-55B3	28C0
28D0	CFCY-018-3B83	CFCY-019-B42E	CFCY-021-45BB	CFCY-022-2507	CFCY-023-7AE2	CFCY-024-10CE	CFCY-025-B011	5007	28D0
28E0	CICY-201-AEBE	CICY-202-83B4	CICY-203-83B4	CICY-204-83B4	CICY-205-A83A	CICY-206-A86A	CICY-207-8CE4	CICY-208-8114	28E0
28F0	CICY-209-AEB2	CICY-210-AD52	CICY-211-AC18	CICY-212-88DA	CICY-213-A064	CICY-214-B2BC	CICY-215-AD42	CICY-216-8E7A	28F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	28-- 29--
2900	CBRC-012-C507	CBRC-013-8185	CBRC-014-E502	CBRC-015-4806	CBRC-016-8A48	CBRC-017-D507	CBRC-018-8185	CBRC-019-F507	2900
2910	CBRC-020-8185	CBRC-023-0040	CBRC-024-55B3	CBRC-025-C49F	CBRC-026-5EA8	CBRC-027-50A0	CBRC-031-2040	CBRC-032-5C2B	2910
2920	CBRC-033-3245	CBRC-034-E1B6	CBRC-035-D428	CBRC-036-224D	CBRC-037-4D25	CBRC-038-2307	CBRC-039-55BB	CBRC-040-72A8	2920
2930	CBRC-041-78A0	CBRC-042-D187	CBRC-043-8185	CBRC-044-5892	CBRC-045-A926	CBRC-005-55B3	CBRC-006-C483	CBRC-007-5EA8	2930
2940	CBRC-008-50A0	CBRC-011-9C03	CLOG-102-6F75	CLOG-103-AB25	CLOG-104-10CE	CLOG-105-5738	CLOG-106-5F10	CLOG-107-67F7	2940
2950	CLOG-108-7718	CLOG-109-25FF	CLOG-110-F4CB	CLOG-111-A848	CLOG-156-AB25	CLOG-157-5F10	CLOG-158-6F51	CLOG-159-A852	2950
2960	CICY-013-B053	CICY-014-B073	CICY-015-A8C4	CICY-016-A8C4	CICY-017-81C0	CICY-018-AE8C	CICY-019-A8A0	CICY-020-A8A0	2960
2970	CICY-021-9044	CICY-022-8108	CICY-023-83B2	CICY-024-83B2	CICY-025-83B0	CICY-026-8104	CICY-029-83B0	CICY-034-8EC8	2970
2980	CMPU-016-8033	CMPU-017-A9B2	CMPU-018-8185	CMPU-019-8185	CLOG-132-6F75	CLOG-133-AB25	CLOG-134-10CE	CLOG-135-5738	2980
2990	CLOG-136-5F10	CLOG-137-67F5	CLOG-138-7718	CLOG-139-25FF	CLOG-140-F48F	CLOG-141-A848	CICY-010-5892	CICY-011-2020	2990
29A0	CICY-012-8185	CMPU-020-81B6	CMPU-021-AA0E	CMPU-022-5B3A	CMPU-023-4B7D	CMPU-024-771A	CMPU-025-351D	CMPU-026-E087	29A0
29B0	CMPU-027-4B75	CMPU-028-771A	CMPU-029-577B	CMPU-030-25FD	CMPU-031-8093	CLOG-004-AB25	CLOG-005-7510	CLOG-006-8185	29B0
29C0	CICY-133-83B4	CICY-134-83B4	CICY-135-83B4	CICY-136-83B4	CICY-137-AE32	CICY-138-A912	CICY-139-ADA8	CICY-140-A93A	29C0
29D0	CICY-141-AA34	CICY-142-AA34	CICY-143-B364	CICY-146-83B4	CICY-164-83B4	CICY-165-83B4	CICY-166-83B4	CICY-167-83B4	29D0
29E0	CICY-169-AC34	CICY-170-AC28	CICY-171-AC0C	CICY-172-AC00	CICY-173-B480	CICY-174-B27A	CICY-175-A57C	CICY-176-B48C	29E0
29F0	CICY-177-AE98	CICY-178-94C2	CICY-179-A4FC	CICY-180-A77C	CICY-181-A06C	CICY-182-8390	CICY-183-AE78	CICY-184-A078	29F0

	0	2	4	6	8	A	C	E	2A--
2A00	CICY-005-A99C	CICY-006-A99C	CICY-007-8446	CICY-008-5498	CICY-009-B463	CMPU-045-5F7D	CMPU-046-A9B2	CMPU-037-5F3A	2A00
2A10	CMPU-038-25FF	CMPU-039-F08B	CMPU-040-5B3A	CMPU-041-4BF3	CMPU-042-7F1A	CMPU-043-351D	CMPU-044-8093	5007	2A10
2A20	CICY-056-2B53	CICY-057-89FC	CSTS-003-AB25	CSTS-004-F933	CSTS-005-0183	CSTS-006-F0AE	CSTS-007-8A26	CSTS-009-AE94	2A20
2A30	CSTS-019-F95E	CSTS-021-ADFC	CSTS-044-F933	CSTS-045-55B3	CSTS-046-5EA8	CSTS-047-AB25	CSTS-048-50AA	CSTS-049-110D	2A30
2A40	CSTS-050-F0AE	CSTS-051-55BB	CSTS-052-3B33	CSTS-053-F055	CSTS-054-57A0	CSTS-055-17F3	CSTS-056-427F	CSTS-057-531C	2A40
2A50	CSTS-058-9248	CSTS-059-8185	CSTS-060-521C	CSTS-061-527F	CSTS-062-17F3	CSTS-063-77A0	CSTS-064-AA50	CSTS-034-70E2	2A50
2A60	CSTS-036-2440	CSTS-037-58E2	CSTS-038-2400	CSTS-039-128E	CSTS-011-AB25	CSTS-012-F933	CSTS-013-5710	CSTS-014-447F	2A60
2A70	CSTS-015-7782	CSTS-016-8185	CICY-003-2C07	CICY-004-8185	CTRT-033-AB25	CTRT-034-14A3	CTRT-035-5569	CTRT-036-25F5	2A70
2A80	CTRT-037-C904	CTRT-038-2555	CTRT-039-5B10	CTRT-040-5BF9	CTRT-041-0F2D	CTRT-042-E096	CTRT-043-F0B9	CTRT-044-0F1B	2A80
2A90	CTRT-045-F0B7	CTRT-046-0F2B	CTRT-047-F0B1	CTRT-048-E01B	CTRT-049-7B10	CTRT-050-5004	CTRT-051-26FF	CTRT-052-F4A4	2A90
2AA0	CTRT-053-5F10	CTRT-054-AA88	CTRT-058-E02D	CTRT-059-E42B	CTRT-060-AA74	CTRT-063-A824	CTRT-061-E428	CTRT-062-A820	2AA0
2AB0	CTRT-055-1427	CTRT-056-3443	CTRT-057-AA98	CTRT-064-1443	CTRT-065-C03F	CTRT-066-A325	CTRT-067-5738	CTRT-068-4755	2AB0
2AC0	CTRT-069-F0E3	CTRT-070-0769	CTRT-071-F4E1	CTRT-072-E05B	CTRT-073-F05A	CTRT-074-7EE2	CTRT-075-5BE9	CTRT-076-2F07	2AC0
2AD0	CTRT-077-2B17	CTRT-078-7FA8	CTRT-079-70A0	CTRT-080-5EB9	CTRT-081-5EE2	CTRT-082-3427	CTRT-083-7518	CTRT-084-AAEA	2AD0
2AE0	CTRT-085-9BDC	CTRT-086-E05D	CTRT-087-7B18	CTRT-088-D06B	CTRT-089-3463	CTRT-090-5773	CTRT-091-0769	CTRT-092-F4F5	2AE0
2AF0	CTRT-093-1483	CTRT-094-AA9C	CTRT-095-F77A	CTRT-096-07FD	CTRT-097-C4FC	CTRT-098-1423	CTRT-099-2707	CTRT-100-3443	2AF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	2A-- 2B--
2B00	CTRT-101-AA9C	BSWI-188-8A54	BSWI-054-DFE6	BSWI-058-2040	BSWI-059-2080	BSWI-060-2406	BSWI-062-E990	BSWI-064-9F36	2B00
2B10	BSWI-068-EB94	BSWI-070-9F68	BSWI-082-2404	BSWI-086-EA9A	BSWI-088-8782	BSWI-100-2482	BSWI-102-FAA0	BSWI-103-B0C4	2B10
2B20	BSWI-107-2400	BSWI-108-2A07	BSWI-110-9248	BSWI-112-E1AB	BSWI-113-8178	BSWI-114-C1E1	BSWI-115-8185	BSWI-132-EBB3	2B20
2B30	BSWI-134-A218	BSWI-144-FBB7	BSWI-148-5007	BSWI-150-EFBB	BSWI-151-A3F2	BSWI-152-DBC9	BSWI-153-DF89	BSWI-162-CBCA	2B30
2B40	BSWI-164-2F07	BSWI-165-508F	BSWI-166-519F	BSWI-167-0602	BSWI-168-AED6	BSWI-169-CF8C	BSWI-170-2404	BSWI-171-FFDA	2B40
2B50	BSWI-172-3BE9	BSWI-173-56AA	BSWI-174-D258	BSWI-175-88B4	BSWI-176-1F2A	BSWI-177-2400	BSWI-178-E903	BSWI-184-8A4C	2B50
2B60	BSWI-116-8C7A	BSWI-117-D1BC	BSWI-118-0040	BSWI-119-FFAF	BSWI-123-59BF	BSWI-124-1BFF	BSWI-129-0640	BSWI-130-2115	2B60
2B70	BSWI-131-8A00	CLOG-126-AB25	CLOG-127-5F10	CLOG-128-6F55	CLOG-129-A852	CNVR-058-2B83	CNVR-059-3B15	CNVR-060-4EA6	2B70
2B80	CNVR-061-5718	CNVR-062-77EF	CNVR-063-E589	CNVR-064-5AE4	CNVR-065-77E8	CNVR-066-0FFB	CNVR-067-F080	CNVR-068-5710	2B80
2B90	CNVR-069-573B	CNVR-070-8F2C	CNVR-071-73AF	CNVR-072-C19A	CNVR-073-9BDC	CNVR-074-73E0	CNVR-075-555B	CNVR-076-3533	2B90
2BA0	CNVR-077-10CE	CNVR-078-E426	CNVR-079-3002	CNVR-080-2407	CNVR-081-2907	CNVR-082-0040	CNVR-083-4EA6	CNVR-084-2207	2BA0
2BB0	CNVR-085-5DE0	CNVR-086-5DC5	CNVR-087-B070	CNVR-088-527B	CNVR-089-0FFB	CNVR-090-F0E8	CNVR-091-77E0	CNVR-092-14FF	2BB0
2BC0	CNVR-093-C4CA	CNVR-094-4203	CNVR-095-790D	CNVR-096-794A	CNVR-097-C14F	CNVR-098-520D	CNVR-099-ABA8	CNVR-110-59F9	2BC0
2BD0	CNVR-111-8176	CNVR-112-C5DA	CNVR-113-E5DC	CNVR-114-CF62	CNVR-115-ABDC	CNVR-116-CF63	CNVR-117-47BB	CNVR-118-0BFB	2BD0
2BE0	CNVR-119-C4E7	CNVR-120-2B93	CNVR-121-89FC	CNVR-122-8185	CNVR-100-5DCD	CNVR-101-B070	CNVR-102-4275	CNVR-103-D1F5	2BE0
2BF0	CNVR-104-C4F8	CNVR-105-5AA4	CNVR-106-77E8	CNVR-107-ABB0	CNVR-108-2040	CNVR-109-ABF4	CDVD-098-8176	CDVD-099-ABE2	2BF0

	0	2	4	6	8	A	C	E	2C--
2C00	CLST-C40-8AC8	CLST-041-C639	CLST-042-B3DA	CLST-043-AC0E	CLST-014-2B83	CLST-015-89FC	CLST-018-8AC8	CLST-019-C625	2C00
2C10	CLST-020-4EE6	CLST-021-4666	CLST-022-D5C7	CLST-023-AC30	CBIN-026-B3F6	CBIN-027-B462	CBIN-028-E4BD	CBIN-029-E5B0	2C10
2C20	CBIN-030-C625	CBIN-031-AC46	CLST-036-2C55	CLST-037-AC48	CLST-030-8AC8	CLST-031-C625	CLST-032-B3DA	CLST-033-C625	2C20
2C30	CLST-034-2C07	CLST-035-AC48	CLST-005-8AC8	CLST-006-C610	CLST-007-B3DA	CLST-008-C610	CLST-009-76A8	CLST-010-7EAA	2C30
2C40	CLST-011-2C75	CLST-012-CD09	CLST-013-8185	CLST-024-2C25	CLST-025-76A8	CLST-026-7EAA	CLST-027-8185	CNVR-005-AB25	2C40
2C50	CNVR-006-0183	CNVR-007-F0D7	CNVR-008-AE94	CNVR-009-9814	CNVR-010-10CE	CNVR-011-F05E	CNVR-012-AB7A	CNVR-013-AE16	2C50
2C60	CNVR-014-2B07	CNVR-015-7A18	CNVR-016-C66A	CNVR-017-B3DA	CNVR-018-2040	CNVR-019-42A6	CNVR-020-44A6	CNVR-021-48A6	2C60
2C70	CNVR-022-2D13	CNVR-023-C679	CNVR-024-AD10	CNVR-025-C674	CNVR-026-ACFE	CNVR-027-C679	CNVR-028-AD00	CNVR-029-AC7A	2C70
2C80	CBRC-119-168D	CBRC-120-C63D	CBRC-121-F03F	CBRC-122-A906	CBRC-091-9816	CBRC-092-55BB	CBRC-093-58A8	CBRC-094-52AA	2C80
2C90	CBRC-095-55B3	CBRC-096-56A8	CBRC-097-5CAA	CBRC-098-6D3B	CBRC-099-6C2D	CBRC-100-679D	CBRC-101-668D	CBRC-102-3B15	2C90
2CA0	CBRC-103-58A8	CBRC-104-52AA	CBRC-105-55BB	CBRC-106-76A8	CBRC-107-7CAA	CBRC-108-0004	CBRC-109-3002	CBRC-110-7D3D	2CA0
2CB0	CBRC-111-7C2D	CBRC-112-779D	CBRC-113-768D	CBRC-114-8178	CBRC-115-E481	CBRC-116-E583	CBRC-117-F007	CBRC-118-8185	2CB0
2CC0	CLST-066-AB25	CLST-067-01E3	CLST-068-F0E4	CLST-069-55BB	CLST-070-5E10	CLST-071-2707	CLST-072-CE50	CLST-073-27F7	2CC0
2CD0	CLST-074-5769	CLST-075-AC48	CLST-055-AB25	CLST-056-01E3	CLST-057-F0E4	CLST-058-55BB	CLST-059-2B2B	CLST-060-52A0	2CD0
2CE0	CLST-061-7210	CLST-062-8185	CLST-063-AE94	CLST-110-AB25	CLST-111-01C3	CLST-112-F0E4	CLST-113-55BB	CLST-114-128E	2CE0
2CF0	CLST-115-5531	CLST-116-6351	CLST-117-C4E3	CLST-118-251D	CLST-119-55BB	CLST-120-128E	5007	CNVR-030-3915	2CF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	2C-- 2D--
2D00	CNVR-031-ED1F	CNVR-032-0002	CNVR-033-799F	CNVR-034-788F	CNVR-035-755F	CNVR-036-744F	CNVR-037-733F	CNVR-038-722F	2D00
2D10	CNVR-039-6FF8	CNVR-040-6EED	CNVR-041-677D	CNVR-042-666D	CNVR-043-ED1F	CNVR-044-2D1B	CNVR-045-128E	CNVR-046-4686	2D10
2D20	CNVR-047-8178	CNVR-048-27CB	CNVR-049-C928	CNVR-050-176B	CNVR-051-D1AC	CNVR-052-3713	CNVR-053-7218	CNVR-054-7418	2D20
2D30	CNVR-055-7618	CNVR-056-8185	CBIN-068-2C25	CBIN-069-8185	CBIN-063-F5BF	CBIN-064-2C55	CBIN-065-8185	CBIN-066-2C75	2D30
2D40	CBIN-067-8185	CBIN-055-B3F6	CBIN-056-B462	CBIN-057-76A8	CBIN-058-7EAA	CBIN-059-E5B9	CBIN-060-F5B5	CBIN-061-2C07	2D40
2D50	CBIN-062-8185	CBIN-008-B3F6	CBIN-009-B460	CBIN-010-E4DA	CBIN-011-168D	CBIN-012-E5CE	CBIN-013-C63B	CBIN-014-2C25	2D50
2D60	CBIN-015-8185	CLOG-162-6F75	CLOG-163-AB25	CLOG-164-10CE	CLOG-165-5738	CLOG-166-5F10	CLOG-167-67F1	CLOG-168-7718	2D60
2D70	CLOG-169-25FF	CLOG-170-F4E9	CLOG-171-A848	CLOG-200-55BB	CLOG-201-2E07	CLOG-202-7EA8	CLOG-203-70A0	CLOG-204-8185	2D70
2D80	CBRC-066-AA06	CBRC-063-2B33	CBRC-064-89FE	CBRC-065-AE94	CBRC-047-AB25	CBRC-048-01E3	CBRC-049-F086	CBRC-050-E182	2D80
2D90	CBRC-051-7CC2	CBRC-052-7892	CBRC-053-74B2	CBRC-054-4806	CBRC-055-0020	CBRC-056-55BB	CBRC-057-C481	CBRC-058-5498	2D90
2DA0	CBRC-059-2B3B	CBRC-060-53A0	CBRC-061-6535	CBRC-062-B463	CBRC-068-0040	CBRC-069-55B3	CBRC-070-C4B5	CBRC-071-5EA8	2DA0
2DB0	CBRC-072-50A0	CBRC-074-2040	CBRC-075-55BB	CBRC-076-56A8	CBRC-077-52AA	CBRC-078-000A	CBRC-079-73A9	CBRC-080-72A9	2DB0
2DC0	CBRC-081-77A9	CBRC-082-76A9	CBRC-083-76A8	CBRC-084-72AA	CBRC-085-D5CD	CBRC-086-8185	CBRC-087-D1CA	CBRC-088-A906	2DC0
2DD0	CTRT-003-AB25	CTRT-004-2607	CTRT-005-2E07	CTRT-006-5F10	CTRT-007-6F3B	CTRT-008-6E2D	CTRT-009-667D	CTRT-010-A225	2DD0
2DE0	CTRT-011-5FF0	CTRT-012-F06D	CTRT-013-7F18	CTRT-014-25FF	CTRT-015-F4D5	CTRT-016-8185	CTRT-017-AB01	CTRT-018-5004	2DE0
2DF0	CTRT-019-25FF	CTRT-020-F4D5	CTRT-021-AA74	CCOM-166-2400	CCOM-167-2C75	CCOM-168-8185	CICY-079-2B23	CICY-080-89FC	2DF0

	0	2	4	6	8	A	C	E	2E--
2EC0	CCOM-025-23F7	CCOM-026-C492	CCOM-027-AE94	CCOM-012-AB25	CCOM-013-01E3	CCOM-014-F084	CCOM-015-5E10	CCOM-016-CE01	2E00
2E10	CCOM-017-2305	CCOM-018-5329	CCOM-019-40E6	CCOM-020-55BB	CCOM-021-56A8	CCOM-022-5EAA	CCOM-023-100E	CCOM-024-128E	2E10
2E20	CLOG-026-6F75	CLOG-027-AB25	CLOG-028-5738	CLOG-029-5F10	CLOG-030-4F7D	CLOG-031-7718	CLOG-032-25FF	CLOG-033-F4A5	2E20
2E30	CLOG-034-8185	CSTS-087-55BB	CSTS-088-5EA0	CSTS-089-4ED3	CSTS-090-5ECB	CSTS-091-1CC5	CSTS-092-FC40	CSTS-093-3C45	2E30
2E40	CSTS-094-7CC2	CSTS-095-8185	CSTS-076-AB25	CSTS-077-2FF7	CSTS-078-5710	CSTS-079-7F10	CSTS-080-C752	CSTS-081-2C55	2E40
2E50	CSTS-082-8185	CSTS-083-2C05	CSTS-084-8185	CLOG-016-6F75	CLOG-017-AB25	CLOG-018-5738	CLOG-019-5F10	CLOG-020-4F7B	2E50
2E60	CLOG-021-7718	CLOG-022-25FF	CLOG-023-F4DB	CLOG-024-8185	CMPU-008-27F5	CMPU-009-C96E	CMPU-010-2755	CMPU-011-5B3A	2E60
2E70	CMPU-012-5BF1	CMPU-013-7F1A	CMPU-014-5559	CMPU-015-8093	CBIN-051-8270	CBIN-052-AD44	CLOG-008-6F75	CLOG-009-AB25	2E70
2E80	CLOG-010-5738	CLOG-011-7718	CLOG-012-25FF	CLOG-013-F481	CLOG-014-8185	CICY-072-B071	CICY-066-A6D8	CICY-067-AB25	2E80
2E90	CICY-068-01C3	CICY-069-F08B	CICY-070-2B63	CICY-071-89FC	CLST-046-55B3	CLST-047-56A8	CLST-048-5EAA	CLST-049-55BB	2E90
2EA0	CLST-050-76A8	CLST-051-7EAA	CLST-052-8185	CLOG-192-AB25	CLOG-193-55BB	CLOG-194-2B3B	CLOG-195-57A0	CLOG-196-7710	2EA0
2EB0	CLOG-197-8185	CLST-086-55BB	CLST-087-5618	CLST-088-5E10	CLST-089-76A8	CLST-090-7EA0	CLST-091-8185	CLST-078-55BB	2EB0
2EC0	CLST-079-56A8	CLST-080-5EAA	CLST-081-7618	CLST-082-7E10	CLST-083-8185	CCOM-162-2400	CCOM-163-2C25	CCOM-164-8185	2EC0
2ED0	CCOM-158-2400	CCOM-159-2C55	CCOM-160-8185	BSTP-002-2404	BSTP-003-CFDF	BSTP-004-DFDF	BSTP-005-CEE0	BSTP-006-AB20	2ED0
2EE0	BSTP-007-81AA	BSTP-008-4E86	BSTP-021-26C3	BSTP-023-2707	BSTP-024-2213	BSTP-025-9ACA	FILE-362-4C52	FILE-363-7C62	2EE0
2EF0	FILE-364-5CC4	FILE-365-7C52	FILE-366-128E	CDVD-046-2C1B	CDVD-047-611D	CDVD-048-600D	CDVD-049-633D	CDVD-050-622D	2EF0

CONTROL STORAGE	0	2	4	6	8	A	C	E	2E-- 2F--
2F00	CDVD-051-ECOD	CDVD-052-699D	CDVD-053-688D	CDVD-054-655D	CDVD-055-644D	CDVD-056-128E	CDVD-057-C418	CDVD-058-1002	2F00
2F10	CDVD-059-79FD	CDVD-060-78ED	CDVD-061-757D	CDVD-062-746D	CDVD-063-5BC9	CDVD-064-52D9	CDVD-065-E827	CDVD-066-C436	2F10
2F20	CDVD-067-968A	CDVD-068-C436	CDVD-069-ABFC	CDVD-070-C42B	CDVD-071-968A	CDVD-072-F839	CDVD-073-968E	CDVD-074-52D9	2F20
2F30	CDVD-075-D5BA	CDVD-076-CD3B	CDVD-077-ABFC	CDVD-078-F82D	CDVD-079-CD25	CDVD-080-5BBB	CDVD-081-72A8	CDVD-082-70AA	2F30
2F40	CDVD-083-1B15	CDVD-084-74A8	CDVD-085-78AA	CDVD-086-8176	CDVD-087-AA76	CMLT-085-5EF9	CMLT-086-53E9	CMLT-087-5239	2F40
2F50	CMLT-088-5129	CMLT-089-5019	CMLT-090-5709	CMLT-091-5679	CMLT-092-5B69	CMLT-093-5AB9	CMLT-094-55A9	CMLT-095-5459	2F50
2F60	CMLT-096-2407	CMLT-097-C56E	CMLT-098-058F	CMLT-099-E0EC	CMLT-100-C16E	CMLT-101-C5EE	CMLT-102-24F7	CMLT-103-128E	2F60
2F70	DCLC-121-1635	DCLC-122-C3F7	DCLC-123-A7AE	DCLC-125-B488	DPTC-059-2185	DPTC-060-213B	DPTC-061-5404	DPTC-062-2413	2F70
2F80	DPTC-063-88BC	DPTC-064-2482	DPTC-065-4F4F	DPTC-066-1D00	DPTC-067-2D04	DPTC-068-2F10	DPTC-069-5B2F	DPTC-070-5F0F	2F80
2F90	DPTC-071-10F3	DPTC-072-EA96	DPTC-073-3013	DPTC-074-2404	DPTC-075-2F43	DPTC-076-5035	DPTC-077-23FD	DPTC-078-0361	2F90
2FA0	DPTC-079-F4A4	DPTC-080-73B1	DPTC-081-88CE	DPTC-082-503D	DPTC-083-2FFF	DPTC-084-FB1D	DPTC-085-5209	DPTC-086-EB1B	2FA0
2FB0	DPTC-087-88CC	DPTC-088-247B	DPTC-089-24AD	DPTC-090-F4BB	DPTC-091-6413	DPTC-092-242B	DPTC-093-25FF	DPTC-094-C4C2	2FB0
2FC0	DPTC-095-A6AE	DPTC-096-054B	DPTC-097-F082	DPTC-098-AF80	DCLB-006-2045	DCLB-007-7022	DCLB-010-2625	DCLB-014-2787	2FC0
2FD0	DCLB-015-81F8	DCLB-017-2486	DCLB-018-2B80	DCLB-019-7212	DCLB-020-24F6	DCLB-021-5622	DCLB-022-2407	DCLB-023-2585	2FD0
2FE0	DCLB-024-3543	DCLB-025-5148	DCLB-026-6171	DCLB-027-C4F7	DCLB-028-C162	DCLB-029-25CB	DCLB-030-F563	DCLB-031-2507	2FE0
2FF0	DCLB-032-AFF8	DCLB-004-C648	DCLB-005-AECA	DCLB-033-5651	DCLB-034-5212	DCLB-035-95F4	CMLT-179-3002	CMLT-180-B472	2FF0

	0	2	4	6	8	A	C	E	30--
3000	DCLR-076-0240	DCLR-002-C3C0	DCLR-003-A7D6	DCLR-013-D623	DCLR-019-1773	DCLR-030-8F7A	DCLR-031-C510	DCLR-032-2EC3	3000
3010	DCLR-033-15C5	DCLR-034-34A5	DCLR-036-C398	DCLR-038-9F5C	DCLR-043-E71E	DCLR-047-F71E	DCLR-048-B478	DCLR-052-B43C	3010
3020	DCLR-053-8D68	DCLR-061-2425	DCLR-062-54E9	DCLR-063-15C5	DCLR-064-1573	DCLR-065-CCAE	DCLR-066-9660	DCLR-067-0D10	3020
3030	DCLR-068-3413	DCLR-069-2205	DCLR-070-7602	DCLR-071-2D40	DCLR-072-C681	DCLR-073-E681	DCLR-074-2B00	DCLR-075-A7A4	3030
3040	DCLR-004-F207	DCLR-008-2400	DCLR-009-2F25	DCLR-010-2E05	DCLR-011-A030	CFCO-120-38E3	CFCO-121-F452	CFCO-122-1B43	3040
3050	CFCO-123-F45F	CFCO-124-56AA	CFCO-125-717D	CFCO-126-706D	CFCO-127-56AA	CFCO-128-737D	CFCO-129-726D	CFCO-130-56AA	3050
3060	CFCO-131-797D	CFCO-132-786D	CFCO-133-56A0	CFCO-134-7F7D	CFCO-135-C1ED	CFCO-136-128E	CFCO-137-7E6D	CFCO-138-128E	3060
3070	CNVR-123-522D	CNVR-124-6223	CNVR-125-5239	CNVR-126-6223	CNVR-127-6223	CNVR-128-6233	CNVR-129-62C3	CNVR-130-128E	3070
3080	DCLE-073-0313	DCLE-074-C4B4	DCLE-075-9ED2	DCLE-048-5F7F	DCLE-049-FC9B	DCLE-050-2007	DCLE-051-4F0F	DCLE-052-2B02	3080
3090	DCLE-053-E618	DCLE-054-FD99	DCLE-055-262B	DCLE-056-FD90	DCLE-057-A78E	DCLE-047-A154	DCLE-060-FCA2	DCLE-061-0D02	3090
30A0	DCLE-062-3483	DCLE-063-5439	DCLE-064-C528	DCLE-065-1565	DCLE-066-5F4F	DCLE-067-FC9B	DCLE-068-E501	DCLE-069-0583	30A0
30B0	DCLE-070-FC81	DCLE-071-3345	DCLE-072-A782	DCLE-040-CD87	DCLE-041-262B	DCLE-042-CD87	DCLE-043-E636	DCLE-044-DD9B	30B0
30C0	DCLE-045-2B00	DCLE-046-0244	DPTQ-003-4642	DPTQ-004-8F74	DPTQ-005-D158	DPTQ-006-C468	DPTQ-007-CED4	DPTQ-008-8F00	30C0
30D0	DPTQ-009-0EC3	DPTQ-010-F0E0	DPTQ-011-D562	DPTQ-012-8D64	DPTQ-013-C465	DPTQ-C14-CAE8	DPTQ-015-3485	DPTQ-016-2E43	30D0
30E0	DPTQ-017-15C5	DPTQ-018-8D68	DPTQ-019-CECF	DPTQ-020-DA63	DPTQ-021-AB20	5007	OVLY-002-3212	OVLY-003-3800	30E0
30F0	OVLY-004-4652	OVLY-006-24C5	OVLY-022-5759	OVLY-024-2CD5	OVLY-C42-57D9	OVLY-C43-4068	OVLY-045-5248	OVLY-055-6021	30F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	30-- 31--
3100	OVLY-056-C48A	OVLY-057-6131	OVLY-058-C48A	OVLY-059-F229	OVLY-060-B0F8	OVLY-062-50C8	OVLY-072-6666	OVLY-073-6201	3100
3110	OVLY-074-C49A	OVLY-075-6311	OVLY-076-C49A	OVLY-077-6068	OVLY-078-B106	OVLY-079-4268	OVLY-080-6021	OVLY-081-C4A4	3110
3120	OVLY-082-6131	OVLY-083-C487	OVLY-084-2820	OVLY-086-2800	OVLY-087-1212	OVLY-088-5007	OVLY-089-B12A	CDMD-151-2040	3120
3130	CDMD-152-4826	CDMD-153-57B9	CDMD-154-C5B8	CDMD-155-0B9F	CDMD-156-008E	CDMD-157-D1CC	CDMD-158-E0E0	CDMD-159-7B1A	3130
3140	CDMD-160-5886	CDMD-161-5710	CDMD-162-0040	CDMD-163-2EFF	CDMD-164-C4B2	CDMD-165-8DB8	CDMD-174-F0BD	CDMD-175-5BBD	3140
3150	CDMD-176-2A93	CDMD-177-56F2	CDMD-178-57C9	CDMD-179-1000	CDMD-180-0BB1	CDMD-181-E0DF	CDMD-182-3002	CDMD-183-9456	3150
3160	CDMD-166-2040	CDMD-167-2A95	CDMD-168-56E2	CDMD-169-57C9	CDMD-170-1000	CDMD-171-0BB9	CDMD-172-F4DD	CDMD-173-9456	3160
3170	CFCO-056-0008	CFCO-057-6FF5	CFCO-058-4886	CFCO-059-F47D	CFCO-060-4226	CFCO-061-4006	CFCO-062-128E	CFHA-006-2C13	3170
3180	CFHA-007-9DE4	CFHA-008-9DE4	CFHA-009-9DE4	CFHA-010-A656	CFHA-011-5CC2	CFHA-012-D5A4	CFHA-013-2A07	CFHA-014-0FF3	3180
3190	CFHA-015-C4A6	CFHA-016-B2F8	CFHA-017-2507	CFHA-018-2EFF	CFHA-019-E4B2	CFHA-020-F4B5	CFHA-021-B18E	CFHA-028-ED27	3190
31A0	CFHA-029-0080	CFHA-030-966C	CFHA-031-2E07	CFHA-032-896A	CFHA-033-817A	CFHA-034-C1AF	CFHA-035-AA76	CFHA-036-57B9	31A0
31B0	CFHA-037-89EC	CFHA-022-F49D	CFHA-023-ED22	CFHA-024-1E8D	CFHA-025-27D3	CFHA-026-2080	CFHA-027-B18E	DUMP-014-2A75	31B0
31C0	DUMP-019-2E13	DUMP-020-25B3	DUMP-021-252D	DUMP-022-2F85	DUMP-023-2F3B	DUMP-C24-C053	DUMP-025-2B45	DUMP-026-A606	31C0
31D0	DUMP-027-A606	DUMP-028-4626	DUMP-029-C05B	DUMP-030-B1E2	DUMP-034-5638	DUMP-035-56B5	DUMP-036-A5FC	DUMP-037-56B9	31D0
31E0	DUMP-038-A5FC	DUMP-039-57B5	DUMP-040-A5FC	DUMP-041-57B9	DUMP-042-A5FC	DUMP-043-2B45	DUMP-044-A606	DUMP-045-053D	31E0
31F0	DUMP-046-C4F4	DUMP-047-A606	DUMP-048-B1F6	DUMP-031-C059	DUMP-032-5628	DUMP-033-B1DA	BWRP-133-2B95	BWRP-134-904C	31F0

	0	2	4	6	8	A	C	E	32--
3200	ETTR-046-3625	ETTR-047-B22E	ETTR-022-EAB4	ETTR-023-5BDF	ETTR-024-5F00	ETTR-025-7D08	ETTR-026-6CF1	ETTR-027-C4A1	3200
3210	ETTR-028-CDA1	ETTR-029-3B00	ETTR-030-2E15	ETTR-031-57FD	ETTR-032-2FFD	ETTR-033-5DE0	ETTR-034-6DC5	ETTR-035-7DE0	3210
3220	ETTR-036-2718	ETTR-037-F0A8	ETTR-038-276B	ETTR-039-ECAB	ETTR-040-0206	ETTR-041-CD81	ETTR-042-DDB0	ETTR-043-2D20	3220
3230	ETTR-044-0D04	ETTR-045-B236	ETTR-048-5BDF	ETTR-049-30B9	ETTR-050-21A5	ETTR-051-2745	ETTR-052-B224	DMCS-076-5D38	3230
3240	DMCS-077-3585	DMCS-078-2B1B	DMCS-079-7BC0	DMCS-080-FE4F	DMCS-081-2D4D	DMCS-082-264D	DMCS-083-F4C4	DMCS-084-CBE7	3240
3250	DMCS-085-D767	DMCS-086-C566	DMCS-090-2515	DMCS-091-C167	DMCS-092-4948	DMCS-093-69D1	DMCS-094-C4D6	DMCS-095-F163	3250
3260	DMCS-096-6446	DMCS-097-4D40	DMCS-098-B244	DMCS-100-5B59	DMCS-101-6511	DMCS-102-C4BE	DMCS-106-2D05	DMCS-107-77C0	3260
3270	DMCS-108-54E2	DMCS-109-56F2	DMCS-110-2105	DMCS-111-2F43	DMCS-112-8D48	CLOG-047-8270	CLOG-048-A86C	5007	3270
3280	CSFT-140-6EA1	CSFT-141-C48C	CSFT-142-D18A	CSFT-143-CF0F	CSFT-144-3000	CSFT-145-CF0E	CSFT-146-3000	CSFT-147-5FE9	3280
3290	CSFT-148-56F9	CSFT-149-5769	CSFT-150-D031	CSFT-151-2707	CSFT-152-128E	CSFT-129-F001	CSFT-130-D026	CSFT-131-5019	3290
32A0	CSFT-132-5309	CSFT-133-5239	CSFT-134-5729	CSFT-135-5679	CSFT-136-5F69	CSFT-137-5EF9	CSFT-138-5AE9	CSFT-139-128E	32A0
32B0	CSFT-153-5279	CSFT-154-5329	CSFT-155-5039	CSFT-156-5109	CSFT-157-2107	CSFT-158-128E	CDVD-006-5618	CDVD-007-5E10	32B0
32C0	CDVD-008-55BB	CDVD-009-F546	CDVD-010-AE94	CDVD-011-9814	CDVD-012-54A8	CDVD-013-58AA	CDVD-014-3B15	CDVD-015-52A8	32C0
32D0	CDVD-016-50AA	CDVD-017-C658	CDVD-018-B3DA	CDVD-019-3B13	CDVD-020-C45E	CDVD-021-9676	CDVD-022-3B23	CDVD-023-4C86	32D0
32E0	CDVD-024-3002	CDVD-025-7DFD	CDVD-026-7CED	CDVD-027-4C46	CDVD-028-7D7D	CDVD-029-7C6D	CDVD-030-F5F0	CDVD-031-ABFC	32E0
32F0	CDVD-032-4C52	CDVD-033-10CE	CDVD-034-AEF8	CDVD-035-8DDC	CFCO-034-5FF3	CFCO-035-48F5	CFCO-036-5883	CFCO-037-4985	32F0

CONTROL STORAGE	0	2	4	6	8	A	C	E	32-- 33--
3300	CFCO-038-5993	CFCO-039-F408	CFCO-040-4595	CFCO-041-128E	CFCO-042-4295	CFCO-043-5223	CFCO-044-4325	CFCO-045-5333	3300
3310	CFCO-046-4035	CFCO-047-5003	CFCO-048-4105	CFCO-049-5113	CFCO-050-4515	CFCO-051-128E	DYPE-202-2791	DYPE-203-5B60	3310
3320	DYPE-204-5BB8	DYPE-205-7B60	DYPE-206-6664	DYPE-207-2404	DYPE-208-8F7A	DYPE-209-0F04	DYPE-210-D139	DYPE-211-C43B	3320
3330	DYPE-212-EEB5	DYPE-213-810C	DYPE-214-8CD4	DYPE-219-8CB8	DYPE-215-8CDE	DYPE-216-3623	DYPE-217-CE37	DYPE-218-8CA4	3330
3340	CSFT-107-D052	CSFT-108-5115	CSFT-109-4013	CSFT-110-5005	CSFT-111-4303	CSFT-112-5335	CSFT-113-4233	CSFT-114-5225	3340
3350	CSFT-115-4723	CSFT-116-5775	CSFT-117-4673	CSFT-118-5665	CSFT-119-4F63	CSFT-120-5FF5	CSFT-121-4EF3	CSFT-122-5EE5	3350
3360	CSFT-123-4AE3	CSFT-124-128E	CSTS-098-55B9	CSTS-104-2125	CSTS-106-2F07	CSTS-107-2178	CSTS-108-8A02	DYPE-072-2F53	3360
3370	DYPE-073-3F15	DYPE-074-1F00	DYPE-075-2F08	DYPE-076-4FFF	DYPE-077-0F20	DYPE-078-3485	DYPE-079-1483	DYPE-080-5DD9	3370
3380	DYPE-081-C487	DYPE-082-3E23	DYPE-083-15C5	DYPE-084-D50D	DYPE-085-3E83	DYPE-086-8FBE	DYPE-087-8378	DYPE-088-AB20	3380
3390	DCLL-002-42A6	DCLL-003-44A6	DCLL-004-2423	DCLL-005-74B8	DCLL-006-70B8	DCLL-007-2465	DCLL-008-74B8	DCLL-009-2115	3390
33A0	DCLL-010-3183	DCLL-011-70B8	DCLL-012-3B45	DCLL-013-46A6	DCLL-014-17FD	DCLL-015-8ADA	DCLL-016-10F5	DCLL-017-24C3	33A0
33B0	DCLL-018-894E	DUMP-002-1F2A	DUMP-003-9812	DUMP-004-2482	DUMP-005-3D02	DUMP-006-2FD3	DUMP-007-0111	DUMP-008-F4C3	33B0
33C0	DUMP-009-211B	DUMP-010-4252	DUMP-011-4426	DUMP-012-A636	CDMD-213-3002	CDMD-214-56F2	CDMD-215-4206	CDMD-216-5D30	33C0
33D0	CDMD-217-5F6A	CDMD-218-7DFF	CDMD-219-7D3A	CDMD-220-C34F	CDMD-221-128E	CCOM-076-2002	CCOM-077-7FFB	CCOM-078-7EEB	33D0
33E0	CCOM-079-777B	CCOM-080-766B	CCOM-081-100A	CCOM-082-128E	ERRQ-101-2A15	ERRQ-102-F772	ERRQ-103-2BF5	ERRQ-104-7CAA	33E0
33F0	ERRQ-105-128E	ERRQ-106-2BB1	ERRQ-107-C4EE	CCOM-045-5218	CCOM-046-5010	CCOM-047-55BB	CCOM-048-56A8	CCOM-049-5EAA	33F0

	0	2	4	6	8	A	C	E	34--
3400	CCOM-050-100E	CCOM-051-128E	DCOM-080-2E07	DCOM-081-7468	DCOM-082-7E68	DCOM-083-7060	DCOM-084-C511	DCOM-085-2A07	3400
3410	DCOM-086-128E	CLST-094-ACE6	CLST-095-52A8	CLST-096-7218	CLST-097-52A0	CLST-098-7218	CLST-099-ACF0	CLST-100-B414	3410
3420	CLST-103-ACE6	CLST-104-5218	CLST-105-72A8	CLST-106-5218	CLST-107-72A0	CLST-108-ACF0	CLST-109-B422	CFCO-090-5EA8	3420
3430	CFCO-091-58A8	CFCO-092-F439	CFCO-093-52A8	CFCO-094-50A0	CFCO-095-1B73	CFCO-096-128E	ERRQ-109-F744	ERRQ-110-D243	3430
3440	ERRQ-111-0D04	ERRQ-112-128E	ERRQ-113-0D80	ERRQ-114-128E	DPTT-088-F04E	DPTT-089-1413	DPTT-090-0010	DPTT-091-4046	3440
3450	DPTT-092-1485	DPTT-093-128E	CFCO-143-5E18	CFCO-144-5818	CFCO-145-F45F	CFCO-146-5218	CFCO-147-5010	CFCO-148-128E	3450
3460	CCOM-086-3002	CCOM-087-7F1D	CCOM-088-7E0D	CCOM-089-773D	CCOM-090-762D	CCOM-091-128E	CMLT-152-3002	CMLT-153-A752	3460
3470	CMLT-154-1002	CMLT-155-3D25	CMLT-156-A752	CMLT-157-89C4	DYPE-237-9816	DYPE-238-B406	DYPE-239-A566	DYPE-240-8792	3470
3480	CLOG-078-8270	CLOG-079-A83C	CFCY-068-817A	CFCY-069-8384	DCLL-019-0214	DCLL-024-B48A	CLOG-144-8270	CLOG-145-8116	3480
3490	DCLR-134-3425	DCLR-135-98D8	5007	5007	5007	5007	5007	5007	3490

	0	2	4	6	8	A	C	E	00--
0000			BSYS-218-0090						0000
0010									0010
0020									0020
0030									0030
0040			ALDP-320-D9C4	ALDP-320-C4C1					0040
0050			ALDP-323-D9C1	ALDP-323-C8C3					0050
0060			ALDP-326-40C4	ALDP-326-C9D3					0060
0070			ALDP-329-C1E5	ALDP-329-D5C9					0070
0080			DCLC-211-0B0C	DCLC-211-0D0E					0080
0090			DCLC-214-0F15	DCLC-214-18FF					0090
00A0									00A0
00B0			BSYS-196-0E0C	BSYS-196-0D1F					00B0
00C0									00C0
00D0									00D0
00E0									00E0
00F0			BSYS-209-0090						00F0

AUX STORAGE	0	2	4	6	8	A	C	E	00-- 01--
0100									0100
0110									0110
0120									0120
0130									0130
0140									0140
0150									0150
0160									0160
0170									0170
0180									0180
0190									0190
01A0									01A0
01B0									01B0
01C0									01C0
01D0									01D0
01E0									01E0
01F0									01F0

	0	2	4	6	8	A	C	E	30--
3000	ERCX-073-40F1	ERCX-073-F2F3	ERCX-073-F4F5	ERCX-073-F6F7	ERCX-073-F061	ERCX-073-E2E3	ERCX-073-E4E5	ERCX-073-E6E7	3000
3010	ERCX-073-60D1	ERCX-073-D2D3	ERCX-073-D4D5	ERCX-074-D6D7	ERCX-074-D0A1	ERCX-074-A2A3	ERCX-074-A4A5	ERCX-074-A6A7	3010
3020	ERCX-074-50C1	ERCX-074-C2C3	ERCX-074-C4C5	ERCX-074-C6C7	ERCX-074-C081	ERCX-074-8283	ERCX-075-8485	ERCX-075-8687	3020
3030	ERCX-075-6A91	ERCX-075-9293	ERCX-075-9495	ERCX-075-9697	ERCX-075-70B1	ERCX-075-B2B3	ERCX-075-B4B5	ERCX-075-B6B7	3030
3040	ERCX-075-F931	ERCX-076-3233	ERCX-076-3435	ERCX-076-3637	ERCX-076-E921	ERCX-076-2223	ERCX-076-2425	ERCX-076-2627	3040
3050	ERCX-076-D911	ERCX-076-1213	ERCX-076-1415	ERCX-076-1617	ERCX-077-A9E1	ERCX-077-6263	ERCX-077-6465	ERCX-077-6667	3050
3060	ERCX-077-C901	ERCX-077-0203	ERCX-077-0405	ERCX-077-0607	ERCX-077-8941	ERCX-077-4243	ERCX-077-4445	ERCX-078-4647	3060
3070	ERCX-078-9951	ERCX-078-5253	ERCX-078-5455	ERCX-078-5657	ERCX-078-B971	ERCX-078-7273	ERCX-078-7475	ERCX-078-7677	3070
3080	ERCX-078-F879	ERCX-078-7A7B	ERCX-079-7C7D	ERCX-079-7E7F	ERCX-079-E869	ERCX-079-E06B	ERCX-079-6C6D	ERCX-079-6E6F	3080
3090	ERCX-079-D859	ERCX-079-5A5B	ERCX-079-5C5D	ERCX-079-5E5F	ERCX-079-A8A0	ERCX-080-AAAB	ERCX-080-ACAD	ERCX-080-AEAF	3090
30A0	ERCX-080-C849	ERCX-080-4A4B	ERCX-080-4C4D	ERCX-080-4E4F	ERCX-080-8880	ERCX-080-8A8B	ERCX-080-8C8D	ERCX-080-8E8F	30A0
30B0	ERCX-081-9890	ERCX-081-9A9B	ERCX-081-9C9D	ERCX-081-9E9F	ERCX-081-B8B0	ERCX-081-BABB	ERCX-081-BCBD	ERCX-081-BEBF	30B0
30C0	ERCX-081-3839	ERCX-081-3A3B	ERCX-081-3C3D	ERCX-082-3E3F	ERCX-082-2829	ERCX-082-2A2B	ERCX-082-2C2D	ERCX-082-2E2F	30C0
30D0	ERCX-082-1819	ERCX-082-1A1B	ERCX-082-1C1D	ERCX-082-1E1F	ERCX-082-6820	ERCX-082-EAEB	ERCX-083-ECED	ERCX-083-EEEF	30D0
30E0	ERCX-083-C809	ERCX-083-0A0B	ERCX-083-0C0D	ERCX-083-0E0F	ERCX-083-4800	ERCX-083-CACB	ERCX-083-CCCD	ERCX-083-CECF	30E0
30F0	ERCX-083-5810	ERCX-084-DADB	ERCX-084-DCDD	ERCX-084-DEDF	ERCX-084-7830	ERCX-084-FAFB	ERCX-084-FCFD	ERCX-084-FEFF	30F0

AUX STORAGE	0	2	4	6	8	A	C	E	30-- 31--
3100									3100
3110									3110
3120									3120
3130									3130
3140									3140
3150									3150
3160									3160
3170									3170
3180									3180
3190									3190
31A0									31A0
31B0									31B0
31C0									31C0
31D0									31D0
31E0									31E0
31F0									31F0

	0	2	4	6	8	A	C	E	50--
5000									5000
5010									5010
5020									5020
5030									5030
5040									5040
5050									5050
5060									5060
5070									5070
5080									5080
5090									5090
50A0									50A0
50B0					DYPE-279-407E	DYPE-279-4C5E	DYPE-279-7A6C	DYPE-279-7D6E	50B0
50C0	DYPE-279-5C4D	DYPE-279-5D7F	DYPE-279-0000	DYPE-279-F07B	DYPE-279-4A6F	DYPE-279-E2E3	DYPE-279-E4E5	DYPE-280-E6E7	50C0
50D0	DYPE-280-E8E9	DYPE-280-6B4F	DYPE-280-7C25	DYPE-280-0000	DYPE-280-6DD1	DYPE-280-D2D3	DYPE-280-D4D5	DYPE-280-D6D7	50D0
50E0	DYPE-280-D8D9	DYPE-280-5B5A	DYPE-281-6015	DYPE-281-1600	DYPE-281-4EC1	DYPE-281-C2C3	DYPE-281-C4C5	DYPE-281-C6C7	50E0
50F0	DYPE-281-C8C9	DYPE-281-4B5F	DYPE-281-5005						50F0

AUX STORAGE	0	2	4	6	8	A	C	E	50-- 51--
5100									5100
5110									5110
5120									5120
5130									5130
5140									5140
5150									5150
5160									5160
5170									5170
5180									5180
5190									5190
51A0									51A0
51B0									51B0
51C0									51C0
51D0									51D0
51E0									51E0
51F0									51F0

	0	2	4	6	8	A	C	E	60--
6000	EPXF-089-B989	EPXF-089-8A8B	EPXF-089-8C8D	EPXF-089-8E8F	EPXF-089-9899	EPXF-089-9A9B	EPXF-089-9C9D	EPXF-089-9E9F	6000
6010	EPXF-089-D949	EPXF-089-4A4B	EPXF-089-4C4D	EPXF-090-4E4F	EPXF-090-5859	EPXF-090-5A5B	EPXF-090-5C5D	EPXF-090-5E5F	6010
6020	EPXF-090-7929	EPXF-090-2A2B	EPXF-090-2C2D	EPXF-090-2E2F	EPXF-090-3839	EPXF-090-3A3B	EPXF-091-3C3D	EPXF-091-3E3F	6020
6030	EPXF-091-F909	EPXF-091-0A0B	EPXF-091-0C0D	EPXF-091-0E0F	EPXF-091-1819	EPXF-091-1A1B	EPXF-091-1C1D	EPXF-091-1E1F	6030
6040	EPXF-091-0CA9	EPXF-092-AAAB	EPXF-092-ACAD	EPXF-092-AEAF	EPXF-092-B891	EPXF-092-9293	EPXF-092-9495	EPXF-092-9697	6040
6050	EPXF-092-80C9	EPXF-092-CACB	EPXF-092-CCCD	EPXF-092-CECF	EPXF-093-D851	EPXF-093-5253	EPXF-093-5455	EPXF-093-5657	6050
6060	EPXF-093-4021	EPXF-093-6A6B	EPXF-093-6C6D	EPXF-093-6E6F	EPXF-093-7831	EPXF-093-C033	EPXF-093-3435	EPXF-094-3637	6060
6070	EPXF-094-E0E9	EPXF-094-EAEB	EPXF-094-ECED	EPXF-094-EEEF	EPXF-094-F811	EPXF-094-1213	EPXF-094-1415	EPXF-094-1617	6070
6080	EPXF-094-B1A1	EPXF-094-A2A3	EPXF-095-A4A5	EPXF-095-A6A7	EPXF-095-B0A8	EPXF-095-B2B3	EPXF-095-B4B5	EPXF-095-B6B7	6080
6090	EPXF-095-D1C1	EPXF-095-C2C3	EPXF-095-C4C5	EPXF-095-C6C7	EPXF-095-D0C8	EPXF-096-D2D3	EPXF-096-D4D5	EPXF-096-D6D7	6090
60A0	EPXF-096-7161	EPXF-096-6263	EPXF-096-6465	EPXF-096-6667	EPXF-096-7068	EPXF-096-7273	EPXF-096-7475	EPXF-096-7677	60A0
60B0	EPXF-097-F1E1	EPXF-097-E2E3	EPXF-097-E4E5	EPXF-097-E6E7	EPXF-097-F0E8	EPXF-097-F2F3	EPXF-097-F4F5	EPXF-097-F6F7	60B0
60C0	EPXF-097-A081	EPXF-097-8283	EPXF-097-8485	EPXF-098-8687	EPXF-098-9088	EPXF-098-BABB	EPXF-098-BCBD	EPXF-098-BEBF	60C0
60D0	EPXF-098-6041	EPXF-098-4243	EPXF-098-4445	EPXF-098-4647	EPXF-098-5048	EPXF-098-DADB	EPXF-099-DCDD	EPXF-099-DEDF	60D0
60E0	EPXF-099-3269	EPXF-099-2223	EPXF-099-2425	EPXF-099-2627	EPXF-099-3028	EPXF-099-7A7B	EPXF-099-7C7D	EPXF-099-7E7F	60E0
60F0	EPXF-099-2021	EPXF-100-0203	EPXF-100-0405	EPXF-100-0607	EPXF-100-1008	EPXF-100-FAFB	EPXF-100-FCFD	EPXF-100-FEFF	60F0

THERE ARE NO DUPLICATE ASSIGNMENTS IN THIS RUN.