

# M8254

(IPBM) FIELD DIAGNOSTIC  
MD-11-DRLPN-A

EP-DRLPN-A-DL  
COPYRIGHT © 1978  
FICHE 1 OF 1

MAR 1978  
**digital**  
MADE IN USA

PRODUCT CODE: MAINDEC-11-DRLPN-A-D  
PRODUCT NAME: M8254 (IPBM) FIELD DIAGNOSTIC  
DATE CREATED: JAN. 1978  
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1978  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.0	LOADING PROCEDURE
3.1	METHOD
3.2	NON-STANDARD ADDRESS, VECTOR, ARE US OF SOFTWARE SWITCH REGISTER
4.0	STARTING PROCEDURE
4.1	CONTROL SWITCH SETTINGS
4.2	STARTING ADDRESS
4.3	PROGRAM AND/OR OPERATOR ACTION
5.0	OPERATING PROCEDURE
5.1	SWITCH REGISTER FUNCTION
5.2	SCOPE LOOPS
5.3	PROGRAM AND/OR OPERATION ACTION
6.0	ERRORS
6.1	ERROR PRINTOUT
7.0	RESTRICTIONS
8.0	MISCELLANEOUS
8.1	POWER FAIL
8.2	XXDP, ACT, APT
8.3	EXECUTION TIME
8.4	LPA-11 (SYSTEM) DIAGNOSTIC SUMMARY

1.0 ABSTRACT  
-----

THIS PROGRAM ALLOWS THE USER TO CHECK-OUT OR DEBUG THE M8254 (IPBM) MODULE. THIS PROGRAM REQUIRES SPECIAL HARDWARE SETUP. THE M8254 MODULE IS PART OF THE LPA-11XX OPTION. THE M8254 JOINS THE UNIBUS AND AN I/O BUS TOGETHER, WITH USE OF TWO MICROPROCESSORS.

REFERENCES TO "MASTER", "SIDE 1", OR "UBSR" REFER TO THE SECTION OF LOGIC ON THE M8254 WHICH RESPONDS OR DEALS WITH PDP-11 UNIBUS. "UBSR" IS THE NAME FOR THE STATUS REGISTER SEEN BY THE UNIBUS SIDE MICRO-PROCESSOR.

REFERENCES TO "SLAVE", "SIDE 2", OR "IOSR" REFER TO THE SECTION OF LOGIC ON THE M8254 WHICH RESPONDS OR DEALS WITH THE I/O BUS MICRO-PROCESSOR. IN THIS DIAGNOSTIC WE CONSIDER THAT SECTION OF BUS ORIGINALLY CONNECTED TO THE M8200-YC TO BE THE IO-BUS.

YOU NEED NOT RUN THIS DIAGNOSTIC UNLESS YOU SUSPECT THE M8254 MODULE IS BAD. YOU SHOULD HAVE ALREADY RUN BOTH M8200-YC DIAGNOSTICS, FOR IF THEY FAIL, THIS DIAGNOSTIC WILL NOT RUN. YOU SHOULD ALSO BE CERTAIN THAT THE KMC-11 IS IN GOOD RUNNING ORDER.

THIS DIAGNOSTIC WILL-NOT-CHECK OUT THE ARBITRATION LOGIC OF THE M8254. TO CHECK ARBITRATION LOGIC RUN "MD-11-DRLPA". IF "NPRS" FAIL, IT WILL REPORT THE PROBLEM AS AN I/O DEVICE ADDRESSING ERROR. IF THE "BR" ARBITRATION CIRCUITRY IS BAD, IT WILL SHOW UP AS A LPA-11 MICRO-CODE FUNCTION ERROR.

TO RUN THIS DIAGNOSTIC, YOU MUST CABLE THE UNIBUS TO THE I/O BUS WITH A UNIBUS CABLE. WHEN DOING THIS, MAKE SURE THE TOTAL UNIBUS LOADS FOR THE SYSTEM DOES NOT EXCEED THE UNIBUS SPEC. ON THE M8200-YC YOU MUST TURN SWITCH 7 OF E76 ON. ALSO W1 MUST BE CUT. SW7 OF E76 DISABLES THE MICRO-CODE FROM RUNNING.

2.0 REQUIREMENTS

2.1 EQUIPMENT

1. PDP-11 FAMILY COMPUTER WITH 16K OF MEMORY (OR MORE) AND CONSOLE I/O FACILITIES (I.E. TTY)
2. M8254 TO BE TESTED
3. 1 KMC-11, 1 M8200-YC
4. 1 UNIBUS CABLE

2.2 STORAGE

THIS PROGRAM OCCUPIES AND USES 16K OF MEMORY.

3.0 LOADING PROCEDURE

## 3.1 METHOD

STANDARD PROCEDURE FOR NORMAL BINARY PROGRAM SHOULD BE FOLLOWED. THIS PROGRAM IS SUPPLIED ON MULTI-MEDIA AND CAN BE LOADED BY XXDP,ACT, OR APT.

## 3.2 NON-STANDARD ADDRESS, VECTOR OR MULTIPLE M8254 CHECK OUT

THIS PROGRAM IS SET UP TO CHECK A M8254 WITH STANDARD TEST STATION SET-UP AS LISTED BELOW. IT IS IMPORTANT THAT IF THE TEST STATION EQUIPMENT ADDRESSES VARY, THAT YOU ONLY CHANGE THESE LOCATIONS. IF YOU RUN MULTIPLE M8254 MODULES, THE ADDRESS VARIANCE BETWEEN TEST SET-UPS IS 20. THIS IS A VARIABLE THAT MAY BE CHANGED. YOU MUST ENTER HOW MANY M8254 MODULES YOU WISH TO RUN, THE DEFAULT AT LOAD TIME IS 1.

<u>TAG</u>	<u>ADDRESS</u>	<u>CONTENTS</u>	<u>COMMENTS</u>
\$BASE	001246	170460	;;BASE ADDRESS OF EQUIPMENT
NOTE			
THIS LOCATION MUST REFLECT THE ADDRESS OF THE KMC-11 USED TO TEST THE FIRST M8254.			
VAR1:	001364	000010	;VARIANCE IN ADDR. BETWEEN ; KMC 1 AND THE M8200-YC.
NOTE			
THIS LOCATION MUST REFLECT THE VARIANCE IN ADDRESS THE M8200-YC HAS OVER THE KMC-11.			
MCNT:	001366	1	;NUMBER OF M8254 MODULES TO ; CHECK OUT
VAR2:	001370	20	;VARIANCE IN ADDR. BETWEEN TEST ; SET-UPS IF TESTING MULTIPLE ; M8254'S.
NOTE			
THIS LOCATION ONLY USED WHEN RUNNING MULTIPLE M8254. IT REFLECTS THE DIFFERENCE IN ADDRESS BETWEEN THE CURRENT TEST STATION KMC-11 AND THE NEXT TEST STATION KMC-11.			
NCNT:	001372	1	;CURRENT TEST STATION

4.0 STARTING PROCEDURE

4.1 CONTROL SWITCH SETTINGS

BEFORE STARTING THE DIAGNOSTIC, SET ALL SWITCH REGISTER BITS AS DESIRED, SEE SECTION 5.1.

4.2 STARTING ADDRESS

200          START OF TEST

4.3 PROGRAM AND/OR OPERATOR ACTION

1. LOAD PROGRAM INTO MEMORY
2. LOAD ADDRESS 200
3. SET SWITCH REGISTER TO DESIRED SETTING
4. START PROGRAM

5.0 OPERATING PROCEDURE

5.1 SWITCH REGISTER FUNCTION

SWR BIT	OCTAL	FUNCTION WHEN SET
15	100000	HALT ON ERROR
14	040000	LOOP ON TEST
13	020000	INHIBIT ERROR TYPEOUT

5.2 SCOPE LOOPS

IF AN ERROR OCCURS AND THE USER WISHES TO SCOPE THE ERROR SWITCH REG. BIT 15 SHOULD BE SET TO HALT ON ERROR. WHEN THE CPU IS HALTED ON ERROR, SWITCH REG BIT 14 (LOOP ON TEST) AND SWR BIT 13 (INHIBIT ERROR TYPEOUT) SHOULD BE SET. SWR BIT 15 SHOULD BE CLEARED AND CPU SHOULD BE CONTINUED.

NOTE

SOME SCOPE LOOPS MAY BE IMPOSSIBLE TO OBTAIN OR NOT REPEATABLE DUE TO THE FACT THAT THREE ASYNCHRONOUS CPUS ARE RUNNING TO GENERATE THE ERROR.

5.3 PROGRAM AND/OR OPERATOR ACTION

1. WHEN THE PROGRAM IS INITIALLY STARTED IT WILL TYPE:

MD-11-DRLPN-A

2. THE PROGRAM EXERCISES THE MB254 ON FIRST PASS.

3. ANY DETECTED ERRORS ARE REPORTED FOR CURRENT PASS.

4. AN "END PASS" MESSAGE IS TYPED.

THE FIRST "PASS" THROUGH THE PROGRAM IS QUICK VERIFY OR A SHORT ONE. ALL OTHER PASSES WILL ITERATE ON EACH SUBTEST UNLESS INHIBITED.



6.0 ERRORS

6.1 ERROR PRINT-OUT

PRINTOUT VARIES WITH THE ERROR DETECTED. THE ERROR PC TYPED OUT IS THE ACTUAL LOCATION OF THE ERROR CALL.

6.2 SET-UP ERRORS

THIS DIAGNOSTIC "WATCHES" THE KMC-11 AND MB200-YC FOR FATAL CONDITIONS THAT WOULD OTHERWISE CAUSE THE DIGNOSTIC TO "HANG". IF THE DIAGNOSTIC EITHER PRINTS AN ERROR OR HALTS INDICATING THIS CONDITION, YOU SHOULD RUN KMC-11 AND/OR MB200-YC DIAGNOSTICS.

7.0 RESTRICTIONS

NONE.

8.0 MISCELLANEOUS  
-----

8.1 POWER FAIL

THIS PROGRAM WILL NOT SUPPORT POWER FAILURES. IF A POWER FAILURE OCCURS, YOU MUST RELOAD AND RESTART THIS PROGRAM.

8.2 XXDP,ACT,APT

THIS PROGRAM IS CHAINABLE UNDER XXDP, ACT, OR APT. ALTHOUGH "APT HOOKS" HAVE BEEN INSTALLED, THEY HAVE NOT BEEN TESTED.

8.3 EXECUTION TIME

THE EXECUTION TIME WILL VARY BETWEEN CPUS. THE APPROXIMATE TIMES ARE LISTED BELOW:

1.0 MINUTE (60 SEC) -NO ERRORS-ITERATIONS INHIBITED  
3.0 MINUTE (180 SEC) -NO ERRORS-WITH ITERATIONS.  
(LISTED UNDER MIS.)

8.4 LPA11 (SYSTEM) DIAGNOSTIC SUMMARY

DIAGNOSTICS FOR THE LPA11 ARE WRITTEN AT THREE LEVELS: (1) TOTAL PDP-11 SYSTEM, (2) LPA11 SYSTEM; AND, (3) LPA11 OPTIONS.

LEVEL 1 IS DESIGNED TO ISOLATE A FAILURE TO THE LPA11 SYSTEM. ALL OPTIONS ON THE PDP-11 ARE EXERCISED.

LEVEL 2 DIAGNOSTICS ISOLATE A FAILURE TO THE INDIVIDUAL OPTION WITHIN THE LPA11. THE LEVEL 2 DIAGNOSTIC IS MD-11-DRLPA. WHEN THE USER RUNS DRLPA HE CAN GENERALLY TELL WHICH OPTION DIAGNOSTIC (LEVEL 3) TO RUN NEXT. M8254 AND M8200-YC ERRORS MAY "LOOK" ALIKE AND DRLPA MAY NOT BE ABLE TO DISTINGUISH BETWEEN THEM. ARBITRATION ERRORS WILL NOT BE DETECTED BY THIS DIAGNOSTIC.

LEVEL THREE DIAGNOSTICS AID IN DETERMINING IF THE ERROR WAS IN FACT ON THE OPTION THE DRLPA SPECIFIED. THE USER MAY "LOOP" ON THE ERROR. WITHIN LEVEL THREE, THERE ARE TWO GROUPS OF DIAGNOSTICS. THE FIRST GROUP REQUIRES NO "EXTRA" WORK BY THE USER IN ORDER TO RUN. GROUP "A" DIAGNOSTICS DO NOT CHECK ARBITRATION, AND REQUIRE EXTRA TIME FOR EXECUTION. THE SECOND GROUP (GROUP "B") REQUIRES THAT THE USER RECONFIGURE THE PDP-11 SYSTEM. THIS RECONFIGURATION INVOLVES CABLING THE UNIBUS TO THE LPA'S I/O BUS.

THE DIAGNOSTIC FOR THE M8254 FALLS INTO THE GROUP "B" CATAGORY.

THE LPA11-KX DIAGNOSTIC KIT WILL INCLUDE:

<u>OPTION</u>	<u>GROUP</u>	<u>DIAG. #</u>	<u>DIAG. TITLE</u>
LPA11-KX	LEVEL 2	MD-11-DRLPA	LPA11-K SYSTEM DIAG.
M8254	"B"	MD-11-DRLPN	M8254 (JPBM) DIAG.
AA11-K	A	MD-11-DRLPB	AA11-K DIAG.
	B	MD-11-DZAAC	AA11-K DIAG.
AR11	A	MD-11-DRLPC	LPA/AR11 DIAG. #1
	A	MD-11-DRLPD	LPA/AR11 DIAG. #2
	A	MD-11-DRLPE	LPA/AR11 DIAG. #3
	B	MD-11-DZARA	AR11 DIAG. #1
	B	MD-11-DZARB	AR11 DIAG. #2
	B	MD-11-DZARC	AR11 DIAG. #3
	B	MD-11-DZALP	AR11 DIAG. #4
DR11-K	A	MD-11-DRLPF	LPA/DR11-K DIAG.
	B	MD-11-DZDRG	DR11-K DIAG.
KW11-K	A	MD-11-DRLPG	LPA/KW11-K DIAG.
	B	MD-11-DZKWK	KW11-K DIAG.
LPS11	A	MD-11-DRLPH	LPA/LPS11 DIAG. #1
	A	MD-11-DRLPI	LPA/LPS11 DIAG. #2
	A	MD-11-DRLPJ	LPA/LPS11 DIAG. #3
	B	MD-11-DZLPC	LPS11 DIAG. #1
	B	MD-11-DZLPD	LPS11 DIAG. #2
	B	MD-11-DZLPI	LPS11 DIAG. #3
AD11-K	A	MD-11-DRLPK	LPA/AD11-K DIAG.
	B	MD-11-DZADL	AD11-K DIAG.
M8200-YC	B	MD-11-DZLPL	LPA/M8200-YC BASIC MICRO-CPU R/W TEST
	B	MD-11-DZLPM	LPA/M8200-YC JMP+ROM READ TEST

16	OPERATIONAL SWITCH SETTINGS
26	TRAP CATCHER
43	BASIC DEFINITIONS
159	ACT11 HOOKS
170	APT PARAMETER BLOCK
192	COMMON TAGS
235	APT MAILBOX-ETABLE
284	ERROR POINTER TABLE
411	INITIALIZE THE COMMON TAGS
482	TYPE PROGRAM NAME
487	GET VALUE FOR SOFTWARE SWITCH REGISTER
508	T1 *TEST THE INITIAL CONDITIONS OF THE IPBM UBSR
547	T2 *TEST THE INITIAL CONDITIONS OF THE IPBM IOSR
575	T3 *TEST THAT INP 17 READS ALL ZEROS ON SIDE 1
615	T4 *TEST THAT INP 17 READS ALL ZEROS ON SIDE 2
653	T5 *TEST THAT INP 13 READS ALL ONES ON SIDE 1
693	T6 *TEST THAT INP 16 READS ALL ONES ON SIDE 2
731	T7 *TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
861	T10 *TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
979	T11 *TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1049	T12 *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1119	T13 *TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1189	T14 *TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1259	T15 *TEST THAT PATTERN 20 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1329	T16 *TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1399	T17 *TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1469	T20 *TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1539	T21 *TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1607	T22 *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1675	T23 *TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1743	T24 *TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1811	T25 *TEST THAT PATTERN 20 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1879	T26 *TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1947	T27 *TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
2015	T30 *TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
2083	T31 *TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA SILO
2139	T32 *TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA SILO
2193	T33 *TEST THAT DATA PATTERN 0 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2227	T34 *TEST THAT DATA PATTERN 1 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2261	T35 *TEST THAT DATA PATTERN 2 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2295	T36 *TEST THAT DATA PATTERN 4 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2329	T37 *TEST THAT DATA PATTERN 10 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2363	T40 *TEST THAT DATA PATTERN 20 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2397	T41 *TEST THAT DATA PATTERN 40 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2431	T42 *TEST THAT DATA PATTERN 100 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2465	T43 *TEST THAT DATA PATTERN 200 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2500	T44 *TEST THAT DATA PATTERN 0 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2533	T45 *TEST THAT DATA PATTERN 1 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2566	T46 *TEST THAT DATA PATTERN 2 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2599	T47 *TEST THAT DATA PATTERN 4 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2632	T50 *TEST THAT DATA PATTERN 10 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2665	T51 *TEST THAT DATA PATTERN 20 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2698	T52 *TEST THAT DATA PATTERN 40 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2731	T53 *TEST THAT DATA PATTERN 100 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1

2764	T54	*TEST THAT DATA PATTERN 200 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2799	T55	*TEST THAT 7/8 FLAG SETS IN SIDE 1 WHEN 56 WORDS WRITTEN IN SILO
2910	T56	*TEST THAT 7/8 FLAG SETS IN SIDE 2 WHEN 56 WORDS WRITTEN IN SILO
3012	T57	*TEST THAT SIDE 1 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRITTEN
3123	T60	*TEST THAT SIDE 2 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRITTEN
3222	T61	*TEST BURST MODE XFERR MASTER TO SLAVE
3299	T62	*TEST BURST MODE XFERR SIDE 2 TO SIDE 1
3368	T63	END OF TESTS
3382		END OF PASS ROUTINE
3785		BINARY TO OCTAL (ASCII) AND TYPE
3862		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3929		ERROR HANDLER ROUTINE
3974		ERROR MESSAGE TIMEOUT ROUTINE
4021		SCOPE HANDLER ROUTINE
4076		TTY INPUT ROUTINE
4216		READ AN OCTAL NUMBER FROM THE TTY
4254		TYPE ROUTINE
4333		APT COMMUNICATIONS ROUTINE
4390		POWER DOWN AND UP ROUTINES
4433		TRAP DECODER
4456		TRAP TABLE

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23  
24  
25  
26  
27  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52

000001  
  
  
  
  
  
  
  
  
  
000000  
  
  
000174 000174  
000176 000000  
  
000200 000200 001464  
000210 000137 012632  
000220 000220 013070  
  
  
  
001100  
  
  
000011  
000012  
000015  
000200  
177776

```

.TITLE MD-11-DRLPN-A
.*COPYRIGHT (C) 1978
.*DIGITAL EQUIPMENT CORP.
.*MAYNARD, MASS. 01754
.*
.*PROGRAM BY EDWARD C. BADGER
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C2), SEPT 14, 1976.
.*
$TN=1

.SBTTL OPERATIONAL SWITCH SETTINGS
.*
.*      SWITCH      USE
.*      -----      -
.*      15          HALT ON ERROR
.*      14          LOOP ON TEST
.*      13          INHIBIT ERROR TYPEOUTS
.*      11          INHIBIT ITERATIONS
.*      10          BELL ON ERROR
.*      8           LOOP ON TEST IN SWR<7:0>
.SBTTL TRAP CATCHER
.*
.*      =0
.*: *ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
.*: *SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
.*: *LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
.*
DISPREG: .WORD 0          ;; SOFTWARE DISPLAY REGISTER
SWREG:   .WORD 0          ;; SOFTWARE SWITCH REGISTER

.*
.*      =200
.*      JMP      START
.*
.*      =210
.*      JMP      MOCRAM
.*
.*      =220
.*      JMP      ODT

.SBTTL BASIC DEFINITIONS
.*: *INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
STACK= 1100
.EQUIV EMT,ERROR          ;; BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE          ;; BASIC DEFINITION OF SCOPE CALL

.*: *MISCELLANEOUS DEFINITIONS
HT= 11                    ;; CODE FOR HORIZONTAL TAB
LF= 12                    ;; CODE FOR LINE FEED
CR= 15                    ;; CODE FOR CARRIAGE RETURN
CRLF= 200                 ;; CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776                ;; PROCESSOR STATUS WORD
.EQUIV PS,PSM

```

```

55      177774      STKLMT= 177774      :: STACK LIMIT REGISTER
56      177772      PIRQ=   177772      :: PROGRAM INTERRUPT REQUEST REGISTER
57      177570      DSWR=   177570      :: HARDWARE SWITCH REGISTER
58      177570      DDISP=  177570      :: HARDWARE DISPLAY REGISTER
59
60      ::*GENERAL PURPOSE REGISTER DEFINITIONS
61      000000      R0=     %0          :: GENERAL REGISTER
62      000001      R1=     %1          :: GENERAL REGISTER
63      000002      R2=     %2          :: GENERAL REGISTER
64      000003      R3=     %3          :: GENERAL REGISTER
65      000004      R4=     %4          :: GENERAL REGISTER
66      000005      R5=     %5          :: GENERAL REGISTER
67      000006      R6=     %6          :: GENERAL REGISTER
68      000007      R7=     %7          :: GENERAL REGISTER
69      000006      SP=     %6          :: STACK POINTER
70      000007      PC=     %7          :: PROGRAM COUNTER
71
72      ::*PRIORITY LEVEL DEFINITIONS
73      000000      PR0=     0           :: PRIORITY LEVEL 0
74      000040      PR1=    40          :: PRIORITY LEVEL 1
75      000100      PR2=   100          :: PRIORITY LEVEL 2
76      000140      PR3=   140          :: PRIORITY LEVEL 3
77      000200      PR4=   200          :: PRIORITY LEVEL 4
78      000240      PR5=   240          :: PRIORITY LEVEL 5
79      000300      PR6=   300          :: PRIORITY LEVEL 6
80      000340      PR7=   340          :: PRIORITY LEVEL 7
81
82      ::*"SWITCH REGISTER" SWITCH DEFINITIONS
83      100000      SW15=  100000        SW15=  100000
84      040000      SW14=   40000        SW14=   40000
85      020000      SW13=  20000        SW13=  20000
86      010000      SW12=  10000        SW12=  10000
87      004000      SW11=   4000        SW11=   4000
88      002000      SW10=  2000         SW10=  2000
89      001000      SW09=  1000         SW09=  1000
90      000400      SW08=   400         SW08=   400
91      000200      SW07=  200          SW07=  200
92      000100      SW06=  100          SW06=  100
93      000040      SW05=   40          SW05=   40
94      000020      SW04=  20           SW04=  20
95      000010      SW03=  10           SW03=  10
96      000004      SW02=   4           SW02=   4
97      000002      SW01=   2           SW01=   2
98      000001      SW00=   1           SW00=   1
99
100     .EQUIV      SW09, SW9
101     .EQUIV      SW08, SW8
102     .EQUIV      SW07, SW7
103     .EQUIV      SW06, SW6
104     .EQUIV      SW05, SW5
105     .EQUIV      SW04, SW4
106     .EQUIV      SW03, SW3
107     .EQUIV      SW02, SW2
108     .EQUIV      SW01, SW1
        .EQUIV      SW00, SW0

```

109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162

100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000200  
000100  
000040  
000020  
000010  
000004  
000002  
000001  
  
000004  
000010  
000014  
000014  
000014  
000020  
000024  
000030  
000034  
000060  
000064  
000240  
  
170460  
000300  
000300  
000001  
  
000224  
000046

.\*DATA BIT DEFINITIONS (BIT00 TO BIT15)

BIT15= 100000  
BIT14= 40000  
BIT13= 20000  
BIT12= 10000  
BIT11= 4000  
BIT10= 2000  
BIT09= 1000  
BIT08= 400  
BIT07= 200  
BIT06= 100  
BIT05= 40  
BIT04= 20  
BIT03= 10  
BIT02= 4  
BIT01= 2  
BIT00= 1  
.EQUIV BIT09,BIT9  
.EQUIV BIT08,BIT8  
.EQUIV BIT07,BIT7  
.EQUIV BIT06,BIT6  
.EQUIV BIT05,BIT5  
.EQUIV BIT04,BIT4  
.EQUIV BIT03,BIT3  
.EQUIV BIT02,BIT2  
.EQUIV BIT01,BIT1  
.EQUIV BIT00,BIT0

.\*BASIC "CPU" TRAP VECTOR ADDRESSES

ERRVEC= 4 ;: TIME OUT AND OTHER ERRORS  
RESVEC= 10 ;: RESERVED AND ILLEGAL INSTRUCTIONS  
TBITVEC= 14 ;: "T" BIT  
TRTVEC= 14 ;: TRACE TRAP  
BPTVEC= 14 ;: BREAKPOINT TRAP (BPT)  
IOTVEC= 20 ;: INPUT/OUTPUT TRAP (IOT) \*\*SCOPE\*\*  
PWRVEC= 24 ;: POWER FAIL  
EMTVEC= 30 ;: EMULATOR TRAP (EMT) \*\*ERROR\*\*  
TRAPVEC= 34 ;: "TRAP" TRAP  
TKVEC= 60 ;: TTY KEYBOARD VECTOR  
TPVEC= 64 ;: TTY PRINTER VECTOR  
PIRQVEC= 240 ;: PROGRAM INTERRUPT REQUEST VECTOR

ABASE= 170460  
AVECT1= 300  
APRIOR= 300  
\$TN=1

.SBTTL ACT11 HOOKS

;;\*\*\*\*\*  
;HOOKS REQUIRED BY ACT11 ;SAVE PC  
\$SVPC=.  
.=46



```

163 000046 011636
164          000052
165 000052 000000
166          000224
167          001000
168
169
170
171
172
173          001000
174          000024
175 000024 000200
176          000044
177 000044 001000
178          001000
179
180
181
182
183 001000
184 001000 000000
185 001002 001172
186 001004 000002
187 001006 000170
188 001010 000170
189 001012 000031

```

```

SENDAD          ;;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOp
.=52
.WORD 0          ;;2)SET LOC.52 TO ZERO
.=SSVPC         ;; RESTORE PC
.=1000
.SBTTL APT PARAMETER BLOCK

;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.SX=.           ;;SAVE CURRENT LOCATION
.=24           ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200           ;;FOR APT START UP
.=44           ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR       ;;POINT TO APT HEADER BLOCK
.=.SX         ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC
;INTERFACE SPEC.

$APTHD:
$HIPTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: .WORD 2 ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 120. ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 120. ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
          .WORD SETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)

```

```

190
191
192
193
194
195
196 001100 001100
197 001100 000000
198 001102 000
199 001103 000
200 001104 000000
201 001106 000000
202 001110 000000
203 001112 000000
204 001114 000
205 001115 001
206 001116 000000
207 001120 000000
208 001122 000000
209 001124 000000
210 001126 000000
211 001130 000000
212 001132 000000
213 001134 000
214 001135 000
215 001136 000000
216 001140 177570
217 001142 177570
218 001144 177560
219 001146 177562
220 001150 177564
221 001152 177566
222 001154 000
223 001155 002
224 001156 012
225 001157 000
226 001160 000000
227 001162 177607
228 001166 077
229 001167 015
230 001170 000012
231
232
233
234
235
236
237 001172
238 001172 000000
239 001174 000000
240 001176 000000
241 001200 000000
242 001202 000000
243 001204 000000

```

000377

.SBTTL COMMON TAGS

```

*****
; THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
; USED IN THE PROGRAM.

```

```

          . =1100
SCMTAG:  .WORD      0          ;; START OF COMMON TAGS
          .BYTE      0          ;; CONTAINS THE TEST NUMBER
SERFLG:  .BYTE      0          ;; CONTAINS ERROR FLAG
SICNT:   .WORD      0          ;; CONTAINS SUBTEST ITERATION COUNT
SLPADR:  .WORD      0          ;; CONTAINS SCOPE LOOP ADDRESS
SLPERR:  .WORD      0          ;; CONTAINS SCOPE RETURN FOR ERRORS
SERITL:  .WORD      0          ;; CONTAINS TOTAL ERRORS DETECTED
SITEMB:  .BYTE      0          ;; CONTAINS ITEM CONTROL BYTE
SERMAX:  .BYTE      1          ;; CONTAINS MAX. ERRORS PER TEST
SERRPC:  .WORD      0          ;; CONTAINS PC OF LAST ERROR INSTRUCTION
SGDADR:  .WORD      0          ;; CONTAINS ADDRESS OF 'GOOD' DATA
SBDADR:  .WORD      0          ;; CONTAINS ADDRESS OF 'BAD' DATA
SGDDAT:  .WORD      0          ;; CONTAINS 'GOOD' DATA
SBDDAT:  .WORD      0          ;; CONTAINS 'BAD' DATA
          .WORD      0          ;; RESERVED--NOT TO BE USED
SAUTOB:  .BYTE      0          ;; AUTOMATIC MODE INDICATOR
SINTAG:  .BYTE      0          ;; INTERRUPT MODE INDICATOR
          .WORD      0
SWR:     .WORD      DSWR       ;; ADDRESS OF SWITCH REGISTER
DISPLAY: .WORD      DDISP     ;; ADDRESS OF DISPLAY REGISTER
STKS:    177560              ;; TTY KBD STATUS
STKB:    177562              ;; TTY KBD BUFFER
STPS:    177564              ;; TTY PRINTER STATUS REG. ADDRESS
STPB:    177566              ;; TTY PRINTER BUFFER REG. ADDRESS
SNUL:    .BYTE      0          ;; CONTAINS NULL CHARACTER FOR FILLS
SFILLS:  .BYTE      2          ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
SFILLC:  .BYTE      12         ;; INSERT FILL CHARS. AFTER A "LINE FEED"
STPFLG:  .BYTE      0          ;; "TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
STIMES:  0                   ;; MAX. NUMBER OF ITERATIONS
SBELL:   .ASCIZ    <207><377><377> ;; CODE FOR BELL
SQUES:   .ASCII    /?/        ;; QUESTION MARK
SCRLF:   .ASCII    <15>       ;; CARRIAGE RETURN
SLF:     .ASCIZ    <12>       ;; LINE FEED

```

.SBTTL APT MAILBOX-ETABLE

```

*****
; EVEN
SMAIL:   .WORD      AMSGTY    ;; APT MAILBOX
MSGTY:   .WORD      AFATAL    ;; MESSAGE TYPE CODE
SFATAL:  .WORD      ATESTN    ;; FATAL ERROR NUMBER
STESTN:  .WORD      APASS     ;; TEST NUMBER
SPASS:   .WORD      ADEVCT    ;; PASS COUNT
SDEVCT:  .WORD      AUNIT     ;; DEVICE COUNT
SUNIT:   .WORD      AUNIT     ;; I/O UNIT NUMBER

```

# F02

MD-11-DRLPN-A      MACY11 27(654) 15-DEC-77 08:43 PAGE 6  
 DRLPN.P11          APT MAILBOX-ETABLE

SEQ 0018

244	001206	000000	\$MSGAD: .WORD	AMSGAD	;; MESSAGE ADDRESS
245	001210	000000	\$MSGLG: .WORD	AMSGLG	;; MESSAGE LENGTH
246	001212		\$ETABLE:		;; APT ENVIRONMENT TABLE
247	001212	000	\$ENV: .BYTE	AENV	;; ENVIRONMENT BYTE
248	001213	000	\$ENVM: .BYTE	AENVM	;; ENVIRONMENT MODE BITS
249	001214	000000	\$SWREG: .WORD	ASWREG	;; APT SWITCH REGISTER
250	001216	000000	\$SUSR: .WORD	AUSR	;; USER SWITCHES
251	001220	000000	\$CPUOP: .WORD	ACPUOP	;; CPU TYPE, OPTIONS
252			.*		BITS 15-11=CPU TYPE
253			.*		11/04=01, 11/05=02, 11/20=03, 11/40=04, 11/45=05
254			.*		11/70=06, PDQ=07, Q=10
255			.*		BIT 10=REAL TIME CLOCK
256			.*		BIT 9=FLOATING POINT PROCESSOR
257			.*		BIT 8=MEMORY MANAGEMENT
258	001222	000	\$MAMS1: .BYTE	AMAMS1	;; HIGH ADDRESS, M.S. BYTE
259	001223	000	\$MTYP1: .BYTE	AMTYP1	;; MEM. TYPE, BLK#1
260			.*		MEM. TYPE BYTE -- (HIGH BYTE)
261			.*		900 NSEC CORE=001
262			.*		300 NSEC BIPOLAR=002
263			.*		500 NSEC MOS=003
264	001224	000000	\$MADR1: .WORD	AMADR1	;; HIGH ADDRESS, BLK#1
265			.*		MEM. LAST ADDR.=3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
266	001226	000	\$MAMS2: .BYTE	AMAMS2	;; HIGH ADDRESS, M.S. BYTE
267	001227	000	\$MTYP2: .BYTE	AMTYP2	;; MEM. TYPE, BLK#2
268	001230	000000	\$MADR2: .WORD	AMADR2	;; MEM. LAST ADDRESS, BLK#2
269	001232	000	\$MAMS3: .BYTE	AMAMS3	;; HIGH ADDRESS, M.S. BYTE
270	001233	000	\$MTYP3: .BYTE	AMTYP3	;; MEM. TYPE, BLK#3
271	001234	000000	\$MADR3: .WORD	AMADR3	;; MEM. LAST ADDRESS, BLK#3
272	001236	000	\$MAMS4: .BYTE	AMAMS4	;; HIGH ADDRESS, M.S. BYTE
273	001237	000	\$MTYP4: .BYTE	AMTYP4	;; MEM. TYPE, BLK#4
274	001240	000000	\$MADR4: .WORD	AMADR4	;; MEM. LAST ADDRESS, BLK#4
275	001242	000300	\$VECT1: .WORD	AVECT1	;; INTERRUPT VECTOR#1, BUS PRIORITY#1
276	001244	000000	\$VECT2: .WORD	AVECT2	;; INTERRUPT VECTOR#2, BUS PRIORITY#2
277	001246	170460	\$BASE: .WORD	ABASE	;; BASE ADDRESS OF EQUIPMENT UNDER TEST
278	001250	000000	\$DEVN: .WORD	ADEVN	;; DEVICE MAP
279	001252	000000	\$CDW1: .WORD	ACDW1	;; CONTROLLER DESCRIPTION WORD#1
280	001254		\$ETEND:		
281			.MEXIT		

282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335

.SBTTL ERROR POINTER TABLE

;\*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.  
;\*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN  
;\*LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.  
;\*NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).  
;\*NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

;\* EM ::POINTS TO THE ERROR MESSAGE  
;\* DH ::POINTS TO THE DATA HEADER  
;\* DT ::POINTS TO THE DATA  
;\* DF ::POINTS TO THE DATA FORMAT

001254

\$ERRTB:

;ITEM 1

EM1  
DH1  
DT1  
DF0

;UBSR ERROR  
;ERRPC GDDAT BAD DATA  
;\$ERRPC,\$GDDAT,\$BDDAT  
;ALL NUMBERS ARE IN OCTAL FORM.

;ITEM 2

EM2  
DH1  
DT1  
DF0

;IOSR ERROR  
;ERRPC GDDAT BAD DATA  
;\$ERRPC,\$GDDAT,\$BDDAT  
;ALL NUMBERS ARE IN OCTAL FORM.

;ITEM 3

EM3  
DH1  
DT1  
DF0

;SIDE 1 (IO TO UB) FAST PATH DATA ERROR  
;ERRPC GDDAT BAD DATA  
;\$ERRPC,\$GDDAT,\$BDDAT  
;ALL NUMBERS ARE IN OCTAL FORM.

;ITEM 4

EM4  
DH1  
DT1  
DF0

;SIDE 2 (UB TO IO) FAST PATH DATA ERROR  
;ERRPC GDDAT BAD DATA  
;\$ERRPC,\$GDDAT,\$BDDAT  
;ALL NUMBERS ARE IN OCTAL FORM.

;ITEM 5

EM5  
DH1  
DT1  
DF0

;SIDE 1 (IO TO UB) SILO DATA ERROR  
;ERRPC GDDAT BAD DATA  
;\$ERRPC,\$GDDAT,\$BDDAT  
;ALL NUMBERS ARE IN OCTAL FORM.

;ITEM 6

EM6

;SIDE 2 (UB TO IO) SILO DATA ERROR

001254 016507  
001256 017102  
001260 017250  
001262 017310  
  
001264 016523  
001266 017102  
001270 017250  
001272 017310  
  
001274 016537  
001276 017102  
001300 017250  
001302 017310  
  
001304 016607  
001306 017102  
001310 017250  
001312 017310  
  
001314 016657  
001316 017102  
001320 017250  
001322 017310  
  
001324 016722

```

336 001326 017102 DH1 ;ERRPC GDDAT BAD DATA
337 001330 017250 DT1 ;SERRPC,$GDDAT,$BDDAT
338 001332 017310 DFO ;ALL NUMBERS ARE IN OCTAL FORM.
339
340 ;ITEM 7
341
342 001334 016765 EM7 ;SIDE 1 INPO DATA ERROR
343 001336 017142 DH7 ;ERRPC INP& # GDDAT BDDAT
344 001340 017262 DT7 ;SERRPC,INNU,$GDDAT,$BDDAT
345 001342 017310 DFO ;ALL NUMBERS ARE IN OCTAL FORM.
346
347 ;ITEM 10
348
349 001344 017015 EM10 ;SIDE 2 INPO DATA ERROR
350 001346 017142 DH7 ;ERRPC INPO # GDDAT BDDAT
351 001350 017262 DT7 ;SERRPC,INNU,$GDDAT,$BDDAT
352 001352 017310 DFO ;ALL NUMBERS ARE IN OCTAL FORM.
353
354 ;ITEM 11
355
356 001354 017045 EM11 ;MICRO-P DETECTED IPBM ERROR
357 001356 017212 DH11 ;TEST ERRPC KMCADR CSR
358 001360 017276 DT11 ;$STSTM,$ERRPC,$DLAD,$BDDAT
359 001362 017310 DFO ;ALL NUMBERS ARE IN OCTAL FORM.
360
361
362
363 001364 000010 VAR1: 10 ;VARIENCE BETWEEN KMC 1 AND M8200-YC
364 001366 000001 MCNT: 1 ;#OF M8254'S TO BE TESTED
365 001370 000020 VAR2: 20 ;VARIENCE IN ADDRS BETWEEN TEST STATIONS
366 001372 000001 NCNT: 1 ;CURRENT TEST STATION
367
368 ;NOTE: TEST STATION CONSISTS OF A
369 ; KMC-11, M8254, AND AN M8200-YC
370
371
372 ;MAINTANCE M8200-YC ADDRESSES
373 001374 170470 KMMCSR: .WORD ABASE+10 ;>PATCH <;M8200-YC BASE ADDRESS
374 001376 170474 KMMADR: .WORD ABASE+14 ;>NO <;M8200-YC CRAM ADDRESS
375 001400 170476 KMMDBR: .WORD ABASE+16 ;>PATCHES <;M8200-YC CRAM DATA
376 001402 170470 MBSELO: .WORD ABASE+10 ;>HERE <BSELO
377 001404 170471 MBSEL1: .WORD ABASE+11 ;>NO <;BSEL1
378 001406 170472 MBSEL2: .WORD ABASE+12 ;>PATCHES <;BSEL2
379 001410 170473 MBSEL3: .WORD ABASE+13 ;>HERE <;BSEL3
380 001412 170474 MBSEL4: .WORD ABASE+14 ;>NO <;BSEL4
381 001414 170475 MBSEL5: .WORD ABASE+15 ;>PATCHES <;BSEL5
382 001416 170476 MBSEL6: .WORD ABASE+16 ;>HERE <;BSEL6
383 001420 170477 MBSEL7: .WORD ABASE+17 ;>NO NO <;BSEL7
384
385 001422 177546 KWADR: .WORD 177546 ;ADDR OF CLOCK.
386
387
388
389 001424 170460 KMCSR: ;1ST KMC-11 ADDRESS
; .WORD ABASE ;>PATCH <;BASE ADDRESS OF KMC-11
    
```

390	001426	170464			KMADR: .WORD	ABASE+4		>ADDRESS	<; CRAM ADDR. REG BITS0:9
391	001430	170466			KMDBR: .WORD	ABASE+6		>"\$BASE"	<; CRAM DATA REG
392	001432	170460			BSELO: .WORD	ABASE		>IF ADDR.	<; BSELO
393	001434	170461			BSEL1: .WORD	ABASE+1		>DIFFERENT	<; BSEL1
394	001436	170462			BSEL2: .WORD	ABASE+2		>	<; BSEL2
395	001440	170463			BSEL3: .WORD	ABASE+3		>NO	<; BSEL3
396	001442	170464			BSEL4: .WORD	ABASE+4		>PATCHES	<; BSEL4
397	001444	170465			BSEL5: .WORD	ABASE+5		>HERE	<; BSEL5
398	001446	170466			BSEL6: .WORD	ABASE+6		>NO	<; BSEL6
399	001450	170467			BSEL7: .WORD	ABASE+7		>PATCHES	<; BSEL7
400									
401	001452	000100			CKVCT: .WORD	100		>VECTOR ADDRESS OF CLOCK.	
402	001454	000102			CKVCT2: .WORD	102			
403	001456	000000			ERCNT: .WORD	0		>TOTAL ERROR COUNT DURING RUN.	
404	001460	000000			INNU: .WORD	0			
405	001462	000000			KLAD: .WORD	0		>ADDR. OF FAILING KMC FOR CERTAIN TESTS.	
406									
407									
408	001464				START:				
409					.SBTTL	INITIALIZE THE COMMON TAGS			
410					;;CLEAR	THE COMMON TAGS (\$CMTAG) AREA			
411	001464	012706	001100		MOV	#\$CMTAG,R6		;;FIRST LOCATION TO BE CLEARED	
412	001470	005026			CLR	(R6)+		;;CLEAR MEMORY LOCATION	
413	001472	022706	001140		CMP	#\$SWR,R6 ; ;DONE?			
414	001476	001374			BNE	-6		;;LOOP BACK IF NO	
415	001500	012706	001100		MOV	#\$STACK,SP		;;SETUP THE STACK POINTER	
416					;;INITIALIZE A FEW VECTORS				
417	001504	012737	014230	000020	MOV	#\$SCOPE,@#IOTVEC		;;IOT VECTOR FOR SCOPE ROUTINE	
418	001512	012737	000340	000022	MOV	#\$340,@#IOTVEC+2		;;LEVEL 7	
419	001520	012737	013726	000030	MOV	#\$ERROR,@#EMTVEC		;;EMT VECTOR FOR ERROR ROUTINE	
420	001526	012737	000340	000032	MOV	#\$340,@#EMTVEC+2		;;LEVEL 7	
421	001534	012737	016226	000034	MOV	#\$TRAP,@#TRAPVEC		;;TRAP VECTOR FOR TRAP CALLS	
422	001542	012737	000340	000036	MOV	#\$340,@#TRAPVEC+2		;;LEVEL 7	
423	001550	012737	016050	000024	MOV	#\$PWARN,@#PWAVEC		;;POWER FAILURE VECTOR	
424	001556	012737	000340	000026	MOV	#\$340,@#PWAVEC+2		;;LEVEL 7	
425	001564	005037	001160		CLR	\$TIMES		;;INITIALIZE NUMBER OF ITERATIONS	
426	001570	012737	001570	001106	MOV	#\$SLPADR		;;INITIALIZE THE LOOP ADDRESS FOR SCOPE	
427					;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS				
428					;;EQUAL TO A "-1" SETUP FOR A SOFTWARE SWITCH REGISTER.				
429	001576	013746	000004		MOV	@#ERRVEC-(SP)		;;SAVE ERROR VECTOR	
430	001602	012737	001636	000004	MOV	#\$64\$,@#ERRVEC		;;SET UP ERROR VECTOR	
431	001610	012737	177570	001140	MOV	#\$DSWR,SWR		;;SETUP FOR A HARDWARE SWICH REGISTER	
432	001616	012737	177570	001142	MOV	#\$DDISP,DISPLAY		;;AND A HARDWARE DISPLAY REGISTER	
433	001624	022777	177777	177306	CMP	#\$-1,@#SWR		;;TRY TO REFERENCE HARDWARE SWR	
434	001632	001012			BNE	66\$		;;BRANCH IF NO TIMEOUT TRAP OCCURRED	
435								;;AND THE HARDWARE SWR IS NOT = -1	
436	001634	000403			BR	65\$		;;BRANCH IF NO TIMEOUT	
437	001636	012716	001644		64\$: MOV	#\$65\$, (SP)		;;SET UP FOR TRAP RETURN	
438	001642	000002			RTI				
439	001644	012737	000176	001140	65\$: MOV	#\$SWREG,SWR		;;POINT TO SOFTWARE SWR	
440	001652	012737	000174	001142	MOV	#\$DISPREG,DISPLAY			
441	001660	012637	000004		66\$: MOV	(SP)+,@#ERRVEC		;;RESTORE ERROR VECTOR	
442									
443	001664	005037	001200		CLR	\$PASS		;;CLEAR PASS COUNT	

```

444 001670 132737 000200 001213 BITB #APTSIZE,$ENVM ;;TEST USER SIZE UNDER APT
445 001676 001403 BEQ 67$ ;;YES,USE NON-APT SWITCH
446 001700 012737 001214 001140 MOV #SSWREG,SWR ;;NO,USE APT SWITCH REGISTER
447 001706 67$: RSTART: CLR ERCNT ;CLEAR ERROR COUNT.
448 001706 005037 001456 RTNAD: MOV $BASE,R0 ;GET BASE ADDRESS.
449 001712 013700 001246 MOV #1,NCNT
450 001716 012737 000001 001372 LSTED: MOV #KMCSR,R1 ;STORAGE
451 001724 012701 001424 MOV RO,(1)+ ;FIX ALL ADDS.
452 001730 010021 MOV #4,R0
453 001732 062700 000004 ADD #2,R0
454 001736 010021 MOV RO,(1)+
455 001740 062700 000002 ADD #2,R0
456 001744 010021 MOV RO,(R1)+
457 001746 013700 001424 MOV KMCSR,R0 ;NO FIX BSEL ADDRESS.
458 001752 012702 000010 MOV #10,R2 ;DO ALL SEVEN.
459 001756 010021 1$: MOV RO,(1)+
460 001760 062700 000001 ADD #1,R0
461 001764 005302 DEC R2
462 001766 001373 BNE 1$
463
464 001770 013737 001424 001374 MOV KMCSR,KMCSR
465 001776 063737 001364 001374 ADD VAR1,KMCSR
466 002004 013737 001374 001376 MOV KMMCSR,KMMADR ;NOW FIX MB200-YC ADDRESS.
467 002012 013737 001374 001400 MOV KMMCSR,KMMDBR
468
469 002020 062737 000004 001376 ADD #4,KMMADR
470 002026 062737 000006 001400 ADD #6,KMMDBR
471 002034 013700 001374 MOV KMCSR,R0 ;NOW FIX BSEL ADDRESSES.
472 002040 012701 001402 MOV #MBSEL0,R1
473 002044 012702 000010 MOV #10,R2
474 002050 010021 2$: MOV RO,(1)+
475 002052 062700 000001 ADD #1,R0
476 002056 005302 DEC R2
477 002060 001373 BNE 2$
478 002062 000005 RESET
479
480 .SBTTL TYPE PROGRAM NAME
481 ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
482 002064 005227 177777 INC #-1 ;;FIRST TIME?
483 002070 001033 BNE 64$ ;;BRANCH IF NO
484 002072 104401 002140 TYPE 65$ ;;TYPE ASCIZ STRING
485 .SBTTL GET VALUE FOR SOFTWARE SWITCH REGISTER
486 002076 005737 000042 TST #42 ;;ARE WE RUNNING UNDER XXDP/ACT?
487 002102 001012 BNE 66$ ;;BRANCH IF YES
488 002104 123727 001212 000001 CMPB $ENV,#1 ;;ARE WE RUNNING UNDER APT?
489 002112 001406 BEQ 66$ ;;BRANCH IF YES
490 002114 023727 001140 000176 CMP SWR,#SWREG ;;SOFTWARE SWITCH REG SELECTED?
491 002122 001005 BNE 67$ ;;BRANCH IF NO
492 002124 104406 GTSWR ;;GET SOFT-SWR SETTINGS
493 002126 000403 BR 67$
494 002130 112737 000001 001134 66$: MOVB #1,$AUTOB ;;SET AUTO-MODE INDICATOR
495 002136 67$: BR 64$ ;;GET OVER THE ASCIZ
496 002136 000410 ;;65$: .ASCIZ <CRLF>#MD-11-DRLPN-A#<CRLF>
497

```

```

498 002160
499 002160 005227 177777
500 002164 001005
501 002166 005227 177777
502 002172 001002
503 002174 104401 016310
504 002200 004537 012246
505
506
507
508
509 002204 000240
510 002206 012737 000001 001160
511 002214 012737 002244 001106
512
513 002222 112737 000001 001102
514 002230 012737 002244 001106
515 002236 012737 002244 001110
516
517 002244
518
519 002244 004537 011656
520 002250 001424
521 000006
522
523 002252 000006
524
525
526 002254 004537 011656
527 002260 001424
528 000001
529
530 002262 000001
531
532
533 002264 117737 177154 001126
534 002272 105037 001127
535 002276 012737 000052 001124
536
537 002304 023737 001124 001126
538 002312 001401

645:
INC #1
BNE 3$ ;IS THIS THE FIRST TIME THROUGH ?
INC #1 ;NO-BRANCH
BNE 3$
TYPE MESWCH
3$: JSR R5,LOAD ;LOAD UCODE INTO KMC.

*****
;TEST 1 *TEST THE INITIAL CONDITIONS OF THE IPBM UBSR
*****
↑ST1: NOP
MOV #1,$TIMES ;DO 1 ITERATION
MOV #1$,SLPADR ;SET SCOPE LOOP ADDRESS
MOVB #1,$STINM ;SET TO TEST #1.
MOV #1$,SLPADR ;LOOP AT 1$.
MOV #1$,SLPERR ;ERROR LOOP AT 1$.

1$:
JSR R5,ISSUEC ;/-MCMD-
.WORD KMCSR ;ISSUE COMMAND TO KMC #1.
.MD.=6
; /READ FAST PATH REG.
JSR R5,ISSUEC ;/-MCMD-
.WORD KMCSR ;ISSUE COMMAND TO KMC #1.
.MD.=1
; /COMMAND=NOP.
JSR R5,ISSUEC ;/-MCMD-
.WORD KMCSR ;ISSUE COMMAND TO KMC #1.
.MD.=1
; /COMMAND=NOP.
; /RETURN HERE AFTER COMMAND .

MOVB @BSEL5,$BDDAT ;READ CSR.
CLRB $BDDAT+1
MOV #52,$GDDAT ;EXPECT BITS 5,3,1 SET, ALL
; OTHER BITS CLEAR.
CMP $GDDAT,$BDDAT ;CSR OK AT INIT?
BEQ TST2
; ***** >> ERROR << *****
ERROR 1 ;#1 KMC SIDE STATUS REG ERROR (UBSR)

*****
;TEST 2 *TEST THE INITIAL CONDITIONS OF THE IPBM IOSR
*****
↑ST2: SCOPE
MOV #1,$TIMES ;DO 1 ITERATION
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
.MD.=1

```



```

552                                     ;/COMMAND=NOP.
553 002332 000001                       .WORD .MD.
554                                     ;/RETURN HERE AFTER COMMAND .
555
556 002334 117737 177054 001126         MOVB  @BSEL5,$BDDAT ;READ IOSR. (IPBM TO SLAVE).
557 002342 105037 001127                 CLR  $BDDAT+1
558 002346 012737 000056 001124         MOV  #56,$GDDAT ;EXPECT BITS 5,3,2,1 SET, ALL OTHER
559                                     ;BITS CLEAR.
560 002354 023737 001124 001126         CMP  $GDDAT,$BDDAT ;SR SIDE 2 (SLAVE) OK AT INIT?
561 002362 001401                         BEQ  TST3
562                                     ;;
563 002364 104002                         ERROR 2
564                                     ;#2 SIDE (IPBM TO SLAVE) STATUS
                                     ;REGISTER ERROR IOSR.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

568
569
570
571                                     ;*****
572                                     ;*TEST 3 *TEST THAT INP 17 READS ALL ZEROS ON SIDE 1
573                                     ;*****
574 002366 000004                         TST3: SCOPE
575
576
577
578 002370 004537 011656                 JSR  R5,ISSUEC ;/-IN1367-
579 002374 001424                         .WORD KMCSR ;/-MCMD-
580 000020                                 .MD.=20 ;/ISSUE COMMAND TO KMC #1.
581
582 002376 000020                         .WORD .MD. ;/READ INPO 17 PUT IN BSEL 3
583                                     ;/RETURN HERE AFTER COMMAND

```

```

584 002400 012737 000017 001460         MOV  #17,INNU ;/RECORD INPO NUMBER.
585 002406 012737 000000 001124         MOV  #0,$GDDAT ;/EXPECT ALL 0Z.
586 002414 005037 001126                 CLR  $BDDAT
587 002420 117737 177014 001126         MOVB  @BSEL3,$BDDAT ;/READ WHAT INP 17 WAS
588 002426 123737 001124 001126         CMPB $GDDAT,$BDDAT ;/WAS IT OK?
589 002434 001401                         BEQ  TST4

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

593 002436 104007                         ERROR 7 ;ERROR INP 17 SHOULD HAVE
594                                     ;BEEN ALL ZEROS
595
596
597
598
599
600
601
602
603
604
605

```

```

NOTE: INP 17 IS AN ADDR. ON THE
CROM BUS. FOR ZEROS, WE WOULD
SELECT A GROUNDED INPUT TO THE
"745151"; FOR ONES, WE WOULD
SELECT A +3 INPUT.
IF ONLY ONE BIT WERE BAD, PROBABLY
THE 745151 WAS BAD. IF MORE THAN
ONE BIT WAS BAD, MAYBE THE
"CROM" SELECTION BIT(S) WERE
BAD, THUS SELECTING A NEW ADDR.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

609
610
611
612
613 002440 000004
614
615
616 002442 004537 012100
617 000020
618
619 002446 000020
620
621 002450 012737 000017 001460
622 002456 012737 000000 001124
623 002464 005037 001126
624 002470 117737 176714 001126
625 002476 123737 001124 001126
626 002504 001401

```

```

;*****
;TEST 4 *TEST THAT INP 17 READS ALL ZEROS ON SIDE 2
;*****
TST4: SCOPE

;--IN1367-
;DMC SIMULATE KMC INSTR. LIST FOR.
JSR RS,KMSIM
.MD.=20
;READ INPO 17 PUT IN BSEL 3
:WORD .MD
;RETURN HERE AFTER COMMAND
MOV #17,INNU ;/RECORD INPO NUMBER.
MOV #0,$GDDAT ;/EXPECT ALL OZ.
CLR $BDDAT
MOVB BMBSEL3,$BDDAT ;/READ WHAT INP 17 WAS
CMPB $GDDAT,$BDDAT ;/WAS IT OK?
BEQ TST5
;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

630 002506 104010
631
632
633
634
635
636
637
638
639
640
641
642

```

```

ERROR 10 ;ERROR INP 17 SHOULD HAVE
;BEEN ALL ZEROS
NOTE: INP 17 IS AN ADDR. ON THE
CROM BUS. FOR ZEROS, WE WOULD
SELECT A GROUNDED INPUT TO THE
"745151"; FOR ONES, WE WOULD
SELECT A +3 INPUT.
IF ONLY ONE BIT WERE BAD, PROBABLY
THE 745151 WAS BAD. IF MORE THAN
ONE BIT WAS BAD, MAYBE THE
"CROM" SELECTION BIT(S) WERE
BAD, THUS SELECTING A NEW ADDR.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

646
647
648
649
650 002510 000004
651
652
653
654 002512 004537 011656
655 002516 001424
656 000016
657
658 002520 000016
659

```

```

;*****
;TEST 5 *TEST THAT INP 13 READS ALL ONES ON SIDE 1
;*****
TST5: SCOPE

;--IN1367-
;--MCMD-
;ISSUE COMMAND TO KMC #1.
JSR RS,ISSUEC
:WORD KMCSR
.MD.=16
;READ INPO 13 PUT IN BSEL3
:WORD .MD
;RETURN HERE AFTER COMMAND .

```

660	002522	012737	000013	001460	MOV	#13, INNU	;/RECORD INPO NUMBER.
661	002530	012737	000377	001124	MOV	#377, \$GDDAT	;/EXPECT ALL OZ.
662	002536	005037	001126		CLR	\$BDDAT	
663	002542	117737	176672	001126	MOV	BSEL3, \$BDDAT	;/READ WHAT INP 13 WAS
664	002550	123737	001124	001126	CMPB	\$GDDAT, \$BDDAT	;/WAS IT OK?
665	002556	001401			BEQ	TST6	;;

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

669	002560	104007			ERROR	7	;	ERROR INP 13 SHOULD HAVE
670							;	BEEN ALL ONES

NOTE: INP 13 IS AN ADDR. ON THE CROM BUS. FOR ZEROS, WE WOULD SELECT A GROUNDED INPUT TO THE "745151"; FOR ONES, WE WOULD SELECT A +3 INPUT. IF ONLY ONE BIT WERE BAD, PROBABLY THE 745151 WAS BAD. IF MORE THAN ONE BIT WAS BAD, MAYBE THE "CROM" SELECTION BIT(S) WERE BAD, THUS SELECTING A NEW ADDR.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

685								
686								
687								
688								
689	002562	000004						
690								
691								
692	002564	004537	012100					
693		000017						
694								
695	002570	000017						
696								
697	002572	012737	000016	001460				
698	002600	012737	000377	001124				
699	002606	005037	001126					
700	002612	117737	176572	001126				
701	002620	123737	001124	001126				
702	002626	001401						

\*\*\*\*\*  
 ;TEST 6 \*TEST THAT INP 16 READS ALL ONES ON SIDE 2  
 \*\*\*\*\*  
 TST6: SCOPE

					JSR	RS, KMSIM	;/-IN1367-
						.MD.=17	;/DMC SIMULATE KMC INSTR. LIST FOR.
							;/READ INPO 16 PUT IN BSEL3
					.WORD	.MD.	
					;/RETURN	HERE AFTER COMMAND	
					MOV	#16, INNU	;/RECORD INPO NUMBER.
					MOV	#377, \$GDDAT	;/EXPECT ALL OZ.
					CLR	\$BDDAT	
					MOV	BSEL3, \$BDDAT	;/READ WHAT INP 16 WAS
					CMPB	\$GDDAT, \$BDDAT	;/WAS IT OK?
					BEQ	TST7	;;

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

706	002630	104010			ERROR	10	;	ERROR INP 16 SHOULD HAVE
707							;	BEEN ALL ONES

NOTE: INP 16 IS AN ADDR. ON THE CROM BUS. FOR ZEROS, WE WOULD SELECT A GROUNDED INPUT TO THE "745151"; FOR ONES, WE WOULD SELECT A +3 INPUT.

708
709
710
711
712
713

714  
715  
716  
717  
718

IF ONLY ONE BIT WERE BAD, PROBABLY  
THE 745151 WAS BAD. IF MORE THAN  
ONE BIT WAS BAD, MAYBE THE  
"CROM" SELECTION BIT(S) WERE  
BAD, THUS SELECTING A NEW ADDR.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

722  
723  
724  
725  
726 002632 000004  
727  
728  
729 002634 004537 012100  
730 000006  
731  
732 002640 000006  
733  
734  
735 002642 112777 000377 176570  
736  
737  
738 002650 004537 011656  
739 002654 001424  
740 000007  
741  
742 002656 000007  
743  
744  
745  
746 002660 004537 011656  
747 002664 001424  
748 000001  
749  
750 002666 000001  
751  
752  
753 002670 117737 176550 001126  
754 002676 105037 001127  
755 002702 012737 000072 001124  
756  
757 002710 023737 001124 001126  
758 002716 001402

\*\*\*\*\*  
\*TEST 7 \*TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG  
\*\*\*\*\*  
†ST7: SCOPE

```
JSR R5,KMSIM ;READ DATA SIDE 2
.MD.=6 ;/DMC SIMULATE KMC INSTR. LIST FOR.
; /READ FAST PATH REG.
.WORD .MD.
;/RETURN HERE AFTER COMMAND .
MOVB #377, @BSEL3 ;SET DATA TO BE WRITTEN
;ISSUE COMMAND TO WRITE IT.
;/-MCMD-
;ISSUE COMMAND TO KMC #1.
JSR R5,ISSUEC
.WORD KMCSR
.MD.=7 ;/WRITE FAST PATH REG.
.WORD .MD.
;/RETURN HERE AFTER COMMAND
;ALLOW SETTLE TIME
;/-MCMD-
;ISSUE COMMAND TO KMC #1.
JSR R5,ISSUEC
.WORD KMCSR
.MD.=1 ;/COMMAND=NOP.
.WORD .MD.
;/RETURN HERE AFTER COMMAND .
MOVB @BSEL5,$BDDAT ;READ SIDE #1 UBSR.
CLRB $BDDAT+1
MOV #72,$GDDAT ;NORMAL=52 EXPECT BIT4
;TO BE SET.
CMP $GDDAT,$BDDAT ;UBSR SIDE 1 OK? (BIT4 SET.)
BEQ 1$
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

762  
763  
764

ERROR 1 ;SIDE 1 CSR BIT4 SHOULD BE SET  
;DATA WRITTEN INTO FAST PATH REG. BUT  
;NOT READ ON SIDE 2.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

MD-11-DRLPN-A  
DRLPN.P11MACY11  
T7

27(654) 15-DEC-77 08:43 PAGE 16

\*TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0028

```

768 002722 000471          BR      TST10          ;;
769
770 002724          1$:      JSR      RS,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
771 002724 004537 012100      .MD.=1          ;/COMMAND=NOP.
772 000001
773
774 002730 000001      .WORD  .MD.
775      ;/RETURN HERE AFTER COMMAND
776 002732 117737 176456 001126  MOVB   @BSEL5,$BDDAT ;READ SIDE #2 IOSR
777 002740 012737 000016 001124  MOV    #16,$GDDAT    ;NORMAL=56, EXPECT BITS
778                                ;TO BE CLEAR.
779 002746 023737 001124 001126  CMP    $GDDAT,$BDDAT ;IS IT CLEAR?
780 002754 001402          BEQ    2$          ;YES-NEXT CHECK.

;*****>> ERROR <<*****

784 002756 104002      ERROR  2          ;BITS OF SIDE #2 IOSR NOT CLEAR WHEN
785                                ;SIDE #1 MADE XFER VIA FAST PATH.

;*****>> ERROR <<*****

789 002760 000452          BR      TST10          ;;
790 002762          2$:      JSR      RS,KMSIM          ;/READ FAST PATH SIDE 2.
791 002762 004537 012100      .MD.=6          ;/DMC SIMULATE KMC INSTR. LIST FOR.
792 000006
793                                ;/READ FAST PATH REG.
794 002766 000006      .WORD  .MD.
795      ;/RETURN HERE AFTER COMMAND
796 002770 117737 176414 001126  MOVB   @BSEL3,$BDDAT ;READ FAST PATH RESULT-ANY DATA
797 002776 001005          BNE    3$          ;GET XFERRED? YES-NEXT CHECK.
798 003000 012737 000377 001124  MOV    #377,$GDDAT  ;HAD EXPECTED 377, BUT WOULD HAVE
799                                ;SETTLED FOR ANYTHING BUT ZERO.

;*****>> ERROR <<*****

803 003006 104003      ERROR  3          ;NO DATA YET THROUGH FAST PATH-
804                                ;SIDE 1 TO SIDE 2.

;*****>> ERROR <<*****

808 003010 000436          BR      TST10          ;;
809
810 003012          3$:      JSR      RS,ISSUEC          ;/-MCMD-
811                                ;/ISSUE COMMAND TO KMC #1.
812 003012 004537 011656      .WORD  KMCSR
813 003016 001424          .MD.=1
814 000001
815                                ;/COMMAND=NOP.
816 003020 000001      .WORD  .MD.
817      ;/RETURN HERE AFTER COMMAND
818 003022 117737 176416 001126  MOVB   @BSEL5,$BDDAT ;READ SIDE 1 UBSR.
819 003030 012737 000052 001124  MOV    #52,$GDDAT   ;EXPECT NORMAL UBSR SETTING.
820 003036 023737 001124 001126  CMP    $GDDAT,$BDDAT ;DID BIT4 CLEAR?
821 003044 001402          BEQ    4$          ;YES-NEXT TEST.

```

```

;*****>> ERROR <<*****
825 003046 104001 ERROR 1 ;TESTING TO SEE THAT BIT4 WOULD CLEAR AFTER
826 ;SIDE 2 FAST PATH REG. WAS READ.

```

```

;*****>> ERROR <<*****
830 003050 000416 BR TST10 ;;
831
832 003052 4S: JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
833 003052 004537 012100 .MD.=1 ;/COMMAND=NOP.
834 000001
835
836 003056 000001 .WORD .MD.
837 ;/RETURN HERE AFTER COMMAND .
838 003060 117737 176330 001126 MOVB @MBSLS,SDDAT ;READ SIDE 2 CSR.
839 003066 012737 000056 001124 MOV #56,$GDDAT ;EXPECT BITS TO SET AFTER FAST
840 ;PATH REG WAS READ.
841 003074 023737 001124 001126 CMP $GDDAT,$SDDAT ;DID BITS SET?
842 003102 001401 BEQ TST10 ;;

```

```

;*****>> ERROR <<*****
846 003104 104000 ERROR ;TESTING THAT SIDE 2 IOSR BITS
847 ;SETS AFTER A FAST PATH REG WAS READ

```

```

;*****>> ERROR <<*****

```

```

851
852 ;*****
853 ;*TEST 10 *TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
854 ;*****
855 003106 000004 †TST10: SCOPE
856
857

```

```

858 003110 004537 011656 JSR R5,ISSUEC ;/-MCM-
859 003114 001424 ;/ISSUE COMMAND TO KMC #1.
860 000006 .WORD KMCSR
861 .MD.=6 ;/READ FAST PATH REG.
862 003116 000006 .WORD .MD.
863 ;/RETURN HERE AFTER COMMAND .
864
865 003120 112777 000377 176262 MOVB #377,@MBSL3 ;SET DATA TO BE WRITTEN
866 ;ISSUE COMMAND TO WRITE IT.
867 003126 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
868 000007 .MD.=7 ;/WRITE FAST PATH REG.
869
870 003132 000007 .WORD .MD.
871 ;/RETURN HERE AFTER COMMAND .
872
873 003134 004537 012100 JSR R5,KMSIM ;ALLOW SETTLE TIME
874 000001 .MD.=1 ;/DMC SIMULATE KMC INSTR. LIST FOR.
875 ;/COMMAND=NOP.

```

# E03

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T10

27(654) 15-DEC-77 08:43 PAGE 18  
\*TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0030

```

876 003140 000001      .WORD .MD.
877                      ;/RETURN HERE AFTER COMMAND .
878
879 003142 117737 176246 001126      MOVB  @MBSEL5,$BDDAT ;READ SIDE #2 IOSR.
880 003150 012737 000076 001124      MOV   #76,$GDDAT    ;NORMAL=56, EXPECT BIT 4
881                      ;TO BE SET.
882 003156 023737 001126 001124      CMP   $BDDAT,$GDDAT ;IOSR SIDE 2 OK? (BIT4 SET).
883 003164 001402 1$                      BEQ   1$

;*****>> ERROR <<*****

887 003166 104002      ERROR 2                ;SIDE 2 IOSR BIT4 SHOULD BE SET
888                      ;DATA WRITTEN INTO FAST PATH REG.
889                      ;BUT NOT READ ON SIDE 1.

;*****>> ERROR <<*****

893 003170 000463      BR    TST11            ;;
894
895 003172 117737 176246 001126 1$:    MOVB  @BSEL5,$BDDAT ;READ SIDE #1 UBSR
896 003200 012737 000012 001124      MOV   #12,$GDDAT   ;NORMAL=51, EXPECT BITS
897                      ;TO BE CLEAR.
898 003206 023737 001124 001126      CMP   $GDDAT,$BDDAT ;IS IT CLEAR?
899 003214 001402 2$                      BEQ   2$            ;YES-NEXT CHECK.

;*****>> ERROR <<*****

903 003216 104001      ERROR 1                ;BITS OF SIDE 1 UBSR NOT CLEAR WHEN
904                      ;SIDE 2 MADE XFER VIA FAST PATH.

;*****>> ERROR <<*****

908 003220 000447      BR    TST11            ;;
909                      ;READ FAST PATH SIDE 1.
910
911 003222                2$:
912
913 003222 004537 011656      JSR   R5,ISSUEC      ;/-MCMD-
914 003226 001424            ;/ISSUE COMMAND TO kMC #1.
915 000006
916                      ;/READ FAST PATH REG.
917 003230 000006      .WORD .MD.
918                      ;/RETURN HERE AFTER COMMAND
919 003232 117737 176202 001126      MOVB  @BSEL3,$BDDAT ;READ FAST PATH. ANY DATA GET
920 003240 001005            ;XFERRED? YES-NEXT CHECK.
921 003242 012737 000377 001124      BNE   3$            ;HAD EXPECTED 377, BUT WOULD HAVE
922                      ;SETTLED FOR ANYTHING BUT ZERO.

;*****>> ERROR <<*****

926 003250 104004      ERROR 4                ;NO DATA GOT THROUGH FAST PATH
927                      ;SIDE 2 TO SIDE 1.

;*****>> ERROR <<*****

```

# F03

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 19  
T10 \*TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0031

```

931 003252 000432          BR      TST11          ;;
932
933 003254          3$:      JSR      R5,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
934 003254 004537 012100      .MD.=1          ;/COMMAND=NOP.
935 000001
936
937 003260 000001      .WORD .MD.
938 ;/RETURN HERE AFTER COMMAND
939 003262 117737 176126 001126      MOVB   @MBSELS,$BDDAT ;READ SIDE 2 IOSR.
940 003270 012737 000056 001124      MOV    #56,$GDDAT    ;EXPECT NORMAL IOSR SETTING
941 003276 023737 001124 001126      CMP    $GDDAT,$BDDAT ;DID BIT4 CLEAR?
942 003304 001402          BEQ    #5             ;YES-NEXT CHECK.

;*****>> ERROR <<*****

946 003306 104002          ERROR  2             ;TESTING TO SEE IF BIT4 WOULD CLEAR AFTER
947                                     ;SIDE 1 FAST PATH REG. WAS READ.

;*****>> ERROR <<*****

951 003310 000413          BR      TST11          ;;
952
953 003312 117737 176126 001126 4$:      MOVB   @MBSELS,$BDDAT ;READ SIDE 1 UBSR.
954 003320 012737 000052 001124      MOV    #52,$GDDAT    ;EXPECT BITS TO SET AFTER FAST PATH
955                                     ;REG. WAS READ.
956 003326 023737 001124 001126      CMP    $GDDAT,$BDDAT ;DID BITS SET?
957 003334 001401          BEQ    TST11          ;;

;*****>> ERROR <<*****

961 003336 104001          ERROR  1             ;TESTING THAT SIDE 1 UBSR BITS
962                                     ;SETS AFTER FAST PATH REG WAS READ.

;*****>> ERROR <<*****

966
967
968
969
970 ;:*****
971 ;*TEST 11          *TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
972 ;*
973 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
974 ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
975 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
976 ;*
977 ;:*****
978 ;TST11: SCOPE
979                                     ;/-DXFRF-
980 003342 004537 011656      JSR    R5,ISSUEC     ;/-MCM-
981 003346 001424                                     ;/ISSUE COMMAND TO KMC #1.
982 000001      .WORD  KMCSR
983      .MD.=1          ;/COMMAND=NOP.

```



G03

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T11

27(654) 15-DEC-77 08:43 PAGE 20

\*TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0032

```

984 003350 000001 .WORD .MD.
985 ;/RETURN HERE AFTER COMMAND .
986
987 003352 012737 000001 001124 MOV #1,$GDDAT ;/RECORD XFERR PATTERN
988 003360 113777 001124 176052 MOVB $GDDAT,$BSEL3 ;/WRITE PATTERN TO U CODE.
989 ;/-MCMD-
990 003366 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
991 003372 001424 .WORD KMCSR
992 000007 .MD.=7
993 ;/WRITE FAST PATH REG.
994 003374 000007 .WORD .MD.
995 ;/RETURN HERE AFTER COMMAND .
996 003376 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
997 000006 .MD.=6
998 ;/READ FAST PATH REG.
999 003402 000006 .WORD .MD.
1000 ;/RETURN HERE AFTER COMMAND .
1001
1002 003404 117737 176000 001126 MOVB @BSEL3,$BDDAT ;/READ SIDE #2
1003 003412 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1004 003420 001401 BEQ 1$

;*****>> ERROR <<*****

1008 003422 104004 ERROR 4 ;/IN PROPER XFER OF PATTERN 1 FROM
1009 ;/SIDE 1 TO SIDE 2

;*****>> ERROR <<*****

1013 003424 005037 001124 1$: CLR $GDDAT ;/NOW XFER A ZERO PATTERN
1014 003430 105077 176004 CLRB @BSEL3
1015 ;/-MCMD-
1016 003434 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1017 003440 001424 .WORD KMCSR
1018 000007 .MD.=7
1019 ;/WRITE FAST PATH REG.
1020 003442 000007 .WORD .MD.
1021 ;/RETURN HERE AFTER COMMAND .
1022 003444 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1023 000006 .MD.=6
1024 ;/READ FAST PATH REG.
1025 003450 000006 .WORD .MD.
1026 ;/RETURN HERE AFTER COMMAND .
1027
1028 003452 117737 175732 001126 MOVB @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1029 003460 001401 BEQ TST12

;*****>> ERROR <<*****

1033 003462 104004 ERROR 4 ;/FAILED TO xFER ZERO PATTERN AFTER

;*****>> ERROR <<*****

1037

```

H03

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T12

27(654) 15-DEC-77 08:43 PAGE 21

\*TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0033

```

1038 ;:*****
1039 ;*TEST 12 *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1040 ;
1041 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1042 ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1043 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1044 ;
1045 ;:*****
1046 003464 000004 ;ST12: SCOPE
1047 ;
1048 ;/DXFRF-
1049 003466 004537 011656 JSR R5,ISSUEC ;/MCMC-
1050 003472 001424 .WORD KMCSR ;/ISSUE COMMAND TO KMC #1.
1051 000001 .MD.=1
1052 ;/COMMAND=NOP.
1053 003474 000001 .WORD .MD.
1054 ;/RETURN HERE AFTER COMMAND .
1055
1056 003476 012737 000002 001124 MOV #2,$GDDAT ;/RECORD XFERR PATTERN
1057 003504 113777 001124 175726 MOVB $GDDAT,$BSEL3 ;/WRITE PATTERN TO U CODE.
1058 ;/MCMC-
1059 003512 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1060 003516 001424 .WORD KMCSR
1061 000007 .MD.=7
1062 ;/WRITE FAST PATH REG.
1063 003520 000007 .WORD .MD.
1064 ;/RETURN HERE AFTER COMMAND .
1065 003522 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1066 000006 .MD.=6
1067 ;/READ FAST PATH REG.
1068 003526 000006 .WORD .MD.
1069 ;/RETURN HERE AFTER COMMAND .
1070
1071 003530 117737 175654 001126 MOVB $BSEL3,$BDDAT ;/READ SIDE #2
1072 003536 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1073 003544 001401 BEQ 1$
;*****>> ERROR <<*****
1077 003546 104004 ERROR 4 ;/IN PROPER XFER OF PATTERN 2 FROM
1078 ;/SIDE 1 TO SIDE 2
;*****>> ERROR <<*****
1082 003550 005037 001124 1$: CLR $GDDAT ;/NOW XFER A ZERO PATTERN
1083 003554 105077 175660 CLRB $BSEL3
1084 ;/MCMC-
1085 003560 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1086 003564 001424 .WORD KMCSR
1087 000007 .MD.=7
1088 ;/WRITE FAST PATH REG.
1089 003566 000007 .WORD .MD.
1090 ;/RETURN HERE AFTER COMMAND .
1091 003570 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.

```

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 22  
T12 \*TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0034

```

1092          000006          .MD.=6
1093          ;/READ FAST PATH REG.
1094 003574 000006          .WORD .MD.
1095          ;/RETURN HERE AFTER COMMAND .
1096
1097 003576 117737 175606 001126  MOVB  @MBSSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1098 003604 001401          BEQ   TST13          ;;
          ;*****>> ERROR <<*****
1102 003606 104004          ERROR  4          ;/FAILED TO XFER ZERO PATTERN AFTER
          ;*****>> ERROR <<*****

```

```

1106
1107          ;*****
1108          ;*TEST 13          *TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
1109          ;*
1110          ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1111          ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1112          ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1113          ;*
1114          ;*****
1115 003610 000004          †TST13: SCOPE
1116          ;/DXFRF-
1117          ;/MCMD-
1118 003612 004537 011656          JSR   R5,ISSUEC          ;/ISSUE COMMAND TO KMC #1.
1119 003616 001424          .WORD KMCSR
1120          000001          .MD.=1
1121          ;/COMMAND=NOP.
1122 003620 000001          .WORD .MD.
1123          ;/RETURN HERE AFTER COMMAND .
1124
1125 003622 012737 000004 001124          MOV   #4,$GDDAT          ;/RECORD XFERR PATTERN
1126 003630 113777 001124 175602          MOVB  $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
1127          ;/MCMD-
1128 003636 004537 011656          JSR   R5,ISSUEC          ;/ISSUE COMMAND TO KMC #1.
1129 003642 001424          .WORD KMCSR
1130          000007          .MD.=7
1131          ;/WRITE FAST PATH REG.
1132 003644 000007          .WORD .MD.
1133          ;/RETURN HERE AFTER COMMAND .
1134 003646 004537 012100          JSR   R5,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
1135          000006          .MD.=6
1136          ;/READ FAST PATH REG.
1137 003652 000006          .WORD .MD.
1138          ;/RETURN HERE AFTER COMMAND .
1139
1140 003654 117737 175530 001126          MOVB  @MBSSEL3,$BDDAT ;/READ SIDE #2
1141 003662 023737 001124 001126          CMP   $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1142 003670 001401          BEQ   IS
          ;*****>> ERROR <<*****

```

J03

MD-11-DRLPN-A MACY11 27(654) 15-DEC-77 08:43 PAGE 23  
DRLPN.P11 T13 \*TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0035

1146 003672 104004 ERROR 4 ;/IN PROPER XFER OF PATTERN 4 FROM  
1147 ;/SIDE 1 TO SIDE 2

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1151 003674 005037 001124 1\$: CLR \$GDDAT ;/NOW XFER A ZERO PATTERN  
1152 003700 105077 175534 CLRB @BSEL3 ;/MCMD-  
1153 ;/ISSUE COMMAND TO KMC #1.  
1154 003704 004537 011656 JSR R5,ISSUEC ;/WRITE FAST PATH REG.  
1155 003710 001424 .WORD KMCSR ;/RETURN HERE AFTER COMMAND  
1156 000007 .MD.=7 ;/DMC SIMULATE KMC INSTR. LIST FOR.  
1157 ;/READ FAST PATH REG.  
1158 003712 000007 .WORD .MD.  
1159 ;/RETURN HERE AFTER COMMAND  
1160 003714 004537 012100 JSR R5,KMSIM ;/READ PATTERN ERROR IF NON-ZERO  
1161 000006 .MD.=6 ;;  
1162 ;/RETURN HERE AFTER COMMAND .  
1163 003720 000006 .WORD .MD.  
1164 ;/RETURN HERE AFTER COMMAND .  
1165 003722 117737 175462 001126 MOVB @MBSEL3,\$BDDAT ;/READ PATTERN ERROR IF NON-ZERO  
1167 003730 001401 BEQ TST14 ;;

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1171 003732 104004 ERROR 4 ;/FAILED TO XFER ZERO PATTERN AFTER  
\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1175  
1176  
1177 ;:\*\*\*\*\*  
1178 ;\*TEST 14 \*TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH RE  
1179 ;\*  
1180 ;\*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1  
1181 ;\*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A  
1182 ;\*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.  
1183 ;\*  
1184 ;:\*\*\*\*\*

1184 003734 000004 †ST14: SCOPE ;/DXFRF-  
1185 ;/MCMD-  
1186 ;/ISSUE COMMAND TO KMC #1.  
1187 003736 004537 011656 JSR R5,ISSUEC ;/COMMAND=NOP.  
1188 003742 001424 .WORD KMCSR ;/RETURN HERE AFTER COMMAND .  
1189 000001 .MD.=1 ;/RECORD XFERR PATTERN  
1190 ;/WRITE PATTERN TO U CODE.  
1191 003744 000001 .WORD .MD.  
1192 ;/MCMD-  
1193 ;/ISSUE COMMAND TO KMC #1.  
1194 003746 012737 000010 001124 MOV #10,\$GDDAT ;/WRITE PATTERN TO U CODE.  
1195 003754 113777 001124 175456 MOVB \$GDDAT,@BSEL3 ;/MCMD-  
1196 ;/ISSUE COMMAND TO KMC #1.  
1197 003762 004537 011656 JSR R5,ISSUEC ;/WRITE PATTERN TO U CODE.  
1198 003766 001424 .WORD KMCSR ;/MCMD-  
1199 000007 .MD.=7 ;/ISSUE COMMAND TO KMC #1.

K03

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T14

27(654) 15-DEC-77 08:43 PAGE 24

\*TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0036

```

1200                                     ;/WRITE FAST PATH REG.
1201 003770 000007 .WORD .MD.
1202 ;/RETURN HERE AFTER COMMAND
1203 003772 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1204 000006 .MD.=6
1205 ;/READ FAST PATH REG.
1206 003776 000006 .WORD .MD.
1207 ;/RETURN HERE AFTER COMMAND .
1208
1209 004000 117737 175404 001126 MOVB @MBSSEL3,$BDDAT ;/READ SIDE #2
1210 004006 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1211 004014 001401 BEQ 15

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1215 004016 104004 ERROR 4 ;/IN PROPER XFER OF PATTERN 10 FROM
1216 ;/SIDE 1 TO SIDE 2

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1220 004020 005037 001124 15: CLR $GDDAT ;/NOW XFER A ZERO PATTERN
1221 004024 105077 175410 CLRB @BSEL3
1222 ;/MCM-
1223 004030 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1224 004034 001424 .WORD KMCSR
1225 000007 .MD.=7

```

;/WRITE FAST PATH REG.

```

1226 004036 000007 .WORD .MD.
1227 ;/RETURN HERE AFTER COMMAND
1228
1229 004040 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1230 000006 .MD.=6

```

;/READ FAST PATH REG.

```

1231 004044 000006 .WORD .MD.
1232 ;/RETURN HERE AFTER COMMAND .
1233
1234

```

```

1235 004046 117737 175336 001126 MOVB @MBSSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1236 004054 001401 BEQ TST15 ;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1240 004056 104004 ERROR 4 ;/FAILED TO XFER ZERO PATTERN AFTER

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1244
1245 ;*****
1246 ;*TEST 15 *TEST THAT PATTERN 20 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH RE
1247 ;*
1248 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1249 ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1250 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1251 ;*
1252 ;*****
1253 004060 000004 TST15: SCOPE

```



# M03

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654)  
T15

15-DEC-77 08:43 PAGE 26

\*TEST THAT PATTERN 20 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0038

```
1309 004202 104004          ERROR 4          ;/FAILED TO XFER ZERO PATTERN AFTER
;*****>> ERROR <<*****
```

```
1313
1314
1315 ;*****
1316 ;*TEST 16          *TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH RE
1317 ;*
1318 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1319 ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1320 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1321 ;*
```

```
1322 004204 000004          ;*****
1323 ;*ST16: SCOPE
1324 ;/DXFRF-
1325 004206 004537 011656   JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1326 004212 001424         .WORD   KMCSR
1327 000001         .MD.=1
1328 ;/COMMAND=NOP.
1329 004214 000001         .WORD   .MD.
1330 ;/RETURN HERE AFTER COMMAND .
1331
1332 004216 012737 000040 001124   MOV      #40,$GDDAT    ;/RECORD XFERR PATTERN
1333 004224 113777 001124 175206   MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO u CODE.
1334 ;/MCMC-
1335 004232 004537 011656   JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1336 004236 001424         .WORD   KMCSR
1337 000007         .MD.=7
1338 ;/WRITE FAST PATH REG.
1339 004240 000007         .WORD   .MD.
1340 ;/RETURN HERE AFTER COMMAND .
1341 004242 004537 012100   JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
1342 000006         .MD.=6
1343 ;/READ FAST PATH REG.
1344 004246 000006         .WORD   .MD.
1345 ;/RETURN HERE AFTER COMMAND .
1346
1347 004250 117737 175134 001126   MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
1348 004256 023737 001124 001126   CMP     $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1349 004264 001401         BEQ     IS
```

```
;*****>> ERROR <<*****
1353 004266 104004          ERROR 4          ;/IN PROPER XFER OF PATTERN 40 FROM
1354 ;/SIDE 1 TO SIDE 2
```

```
;*****>> ERROR <<*****
1358 004270 005037 001124          1S: CLR      $GDDAT      ;/NOW XFER A ZERO PATTERN
1359 004274 105077 175140          CLRB    @BSEL3
1360 ;/MCMC-
1361 004300 004537 011656          JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
```

# N03

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T16

27(654) 15-DEC-77 08:43 PAGE 27

\*TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG

SEQ 0039

```

1362 004304 001424 .WORD KMCSR
1363 000007 .MD.=7
1364 ;/WRITE FAST PATH REG.
1365 004306 000007 .WORD .MD.
1366 ;/RETURN HERE AFTER COMMAND
1367 004310 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1368 000006 .MD.=6
1369 ;/READ FAST PATH REG.
1370 004314 000006 .WORD .MD.
1371 ;/RETURN HERE AFTER COMMAND .
1372
1373 004316 117737 175066 001126 MOVB @MSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1374 004324 001401 BEQ TST17 ;

```

```

;*****>> ERROR <<*****
1378 004326 104004 ERROR 4 ;/FAILED TO XFER ZERO PATTERN AFTER
;*****>> ERROR <<*****

```

```

1382
1383 ;*****
1384 ;*TEST 17 *TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH R
1385 ;*
1386 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1387 ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1388 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1389 ;*

```

```

1390 ;*****
1391 004330 000004 TST17: SCOPE
1392 ;/DXFRF-
1393 ;/MCMD-
1394 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1395 004332 004537 011656 .WORD KMCSR
1396 004336 001424 .MD.=1
1397 ;/COMMAND=NOP.
1398 004340 000001 .WORD .MD.
1399 ;/RETURN HERE AFTER COMMAND .
1400
1401 004342 012737 000100 001124 MOV #100,$GDDAT ;/RECORD XFERR PATTERN
1402 004350 113777 001124 175062 MOVB $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
1403 ;/MCMD-
1404 004356 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO kMC #1.
1405 004362 001424 .WORD KMCSR
1406 000007 .MD.=7
1407 ;/WRITE FAST PATH REG.
1408 004364 000007 .WORD .MD.
1409 ;/RETURN HERE AFTER COMMAND
1410 004366 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1411 000006 .MD.=6
1412 ;/READ FAST PATH REG.
1413 004372 000006 .WORD .MD.
1414 ;/RETURN HERE AFTER COMMAND .
1415

```



1416 004374 117737 175010 001126  
1417 004402 023737 001124 001126  
1418 004410 001401

MOVB @MSEL3,\$BDDAT ;/READ SIDE #2  
CMP \$GDDAT,\$BDDAT ;/DATA SENT=DATA RECEIVED?  
BEQ 1\$

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1422 004412 104004  
1423

ERROR 4 ;/IN PROPER XFER OF PATTERN 100 FROM  
;/SIDE 1 TO SIDE 2

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1427 004414 005037 001124  
1428 004420 105077 175014

1\$:

CLR \$GDDAT ;/NOW XFER A ZERO PATTERN  
CLRB @BSEL3

1429  
1430 004424 004537 011656  
1431 004430 001424  
1432 000007

;/-MCMD-  
JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=7

1433  
1434 004432 000007

;/WRITE FAST PATH REG.

1435  
1436 004434 004537 012100  
1437 000006

.WORD .MD.  
;/RETURN HERE AFTER COMMAND  
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=6

1438  
1439 004440 000006

;/READ FAST PATH REG.

1440  
1441  
1442 004442 117737 174742 001126  
1443 004450 001401

.WORD .MD.  
;/RETURN HERE AFTER COMMAND .

MOVB @MSEL3,\$BDDAT ;/READ PATTERN ERROR IF NON-ZERO  
BEQ TST20 ;;

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1447 004452 104004

ERROR 4 ;/FAILED TO XFER ZERO PATTERN AFTER

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1451  
1452  
1453  
1454  
1455  
1456  
1457  
1458  
1459  
1460 004454 000004  
1461  
1462  
1463 004456 004537 011656  
1464 004462 001424  
1465 000001  
1466  
1467 004464 000001  
1468  
1469

\*\*\*\*\*  
; \*TEST 20 \*TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH R

;\*  
;\*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1  
;\*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A  
;\*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.  
;\*  
\*\*\*\*\*

\*\*\*\*\*  
†ST20: SCOPE

;/-DXFRF-  
;/-MCMD-  
JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=1

;/COMMAND=NOP.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND .

```

1470 004466 012737 000200 001124      MOV      #200,$GDDAT      ;/RECORD XFERR PATTERN
1471 004474 113777 001124 174736      MOVVB   $GDDAT,$BSEL3 ;/WRITE PATTERN TO u CODE.
1472                                     ;/-MCMD-
1473 004502 004537 011656      JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1474 004506 001424                                     .WORD   KMCSR
1475 000007                                     .MD.=7
1476                                     ;/WRITE FAST PATH REG.
1477 004510 000007      .WORD   .MD.
1478                                     ;/RETURN HERE AFTER COMMAND
1479 004512 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
1480 000006                                     .MD.=6
1481                                     ;/READ FAST PATH REG.
1482 004516 000006      .WORD   .MD.
1483                                     ;/RETURN HERE AFTER COMMAND
1484
1485 004520 117737 174664 001126      MOVVB   @BSEL3,$BDDAT ;/READ SIDE #2
1486 004526 023737 001124 001126      CMP     $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1487 004534 001401      BEQ     1$

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1491 004536 104004      ERROR   4                ;/IN PROPER XFER OF PATTERN 200 FROM
1492                                     ;/SIDE 1 TO SIDE 2

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1496 004540 005037 001124      1$: CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
1497 004544 105077 174670      CLRVB  @BSEL3
1498                                     ;/-MCMD-
1499 004550 004537 011656      JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1500 004554 001424                                     .WORD   KMCSR
1501 000007                                     .MD.=7
1502                                     ;/WRITE FAST PATH REG.
1503 004556 000007      .WORD   .MD.
1504                                     ;/RETURN HERE AFTER COMMAND
1505 004560 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
1506 000006                                     .MD.=6
1507                                     ;/READ FAST PATH REG.
1508 004564 000006      .WORD   .MD.
1509                                     ;/RETURN HERE AFTER COMMAND
1510
1511 004566 117737 174616 001126      MOVVB   @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1512 004574 001401      BEQ     TST21           ;;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1516 004576 104004      ERROR   4                ;/FAILED TO xFER ZERO PATTERN AFTER

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1520  
1521  
1522  
1523

```

;*****
; *TEST 21 *TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
; *

```



# E04

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T21

27(654) 15-DEC-77 08:43 PAGE 31

\*TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0043

```
1578 004710 117737 174524 001126      MOVB  @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1579 004716 001401                      BEQ   TST22          ;;
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
1583 004720 104003                      ERROR  3              ;FAILED TO XFER ZERO PATTERN.
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596 004722 000004
1597
1598 004724 004537 012100
1599 000001
1600
1601 004730 000001
1602
1603
1604 004732 012737 000002 001124
1605 004740 113777 001124 174442
1606 004746 004537 012100
1607 000007
1608
1609 004752 000007
1610
1611
1612 004754 004537 011656
1613 004760 001424
1614 000006
1615
1616 004762 000006
1617
1618
1619 004764 117737 174450 001126
1620 004772 023737 001124 001126
1621 005000 001401
```

```
::*****
;*TEST 22      *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
```

```
;*
; *IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
; *TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
; *PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
;*
```

```
::*****
```

```
†TST22: SCOPE
```

```
;/-DXFRF-
; /DMC SIMULATE KMC INSTR. LIST FOR.
.MD.=1
; /COMMAND=NOP.
```

```
.WORD .MD.
; /RETURN HERE AFTER COMMAND .
```

```
MOV  #2,$GDDAT ;/RECORD XFERR PATTERN
MOV  $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
JSR  R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
.MD.=7
; /WRITE FAST PATH REG.
```

```
.WORD .MD.
; /RETURN HERE AFTER COMMAND .
;/-MCMD-
; /ISSUE COMMAND TO KMC #1.
```

```
.WORD KMCSR
.MD.=6
; /READ FAST PATH REG.
```

```
.WORD .MD.
; /RETURN HERE AFTER COMMAND .
```

```
MOV  @BSEL3,$BDDAT ;/READ SIDE #1
CMP  $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
BEQ  1$
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
1625 005002 104003                      ERROR  3              ;INPROPER XFER OF PATTERN 2 FROM
1626                                     ;/SIDE 2 TO SIDE 1
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
1630 005004 005037 001124
1631 005010 105077 174374
```

```
1$: CLR  $GDDAT ;/NOW XFER A ZERO PATTERN
    CLRB @BSEL3
```

# F04

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T22

27(654) 15-DEC-77 08:43 PAGE 32

\*TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0044

```

1632 005014 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1633 000007 .MD.=7 ;/WRITE FAST PATH REG.
1634 ;/RETURN HERE AFTER COMMAND
1635 005020 000007 .WORD .MD. ;/MCMC-
1636 ;/ISSUE COMMAND TO KMC #1.
1637 ;/READ FAST PATH REG.
1638 005022 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1639 005026 001424 .WORD KMCSR
1640 000006 .MD.=6
1641 ;/READ FAST PATH REG.
1642 005030 000006 .WORD .MD.
1643 ;/RETURN HERE AFTER COMMAND .
1644
1645 005032 117737 174402 001126 MOVB @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1646 005040 001401 BEQ TST23 ;
;*****>> ERROR <<*****
1650 005042 104003 ERROR 3 ;FAILED TO XFER ZERO PATTERN.
;*****>> ERROR <<*****

```

```

1654
1655 ;:*****
1656 ;*TEST 23 *TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1657 ;*
1658 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
1659 ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
1660 ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1661 ;*
1662 ;:*****
1663 005044 000004 †TST23: SCOPE ;/DXFRF-
1664 ;/DMC SIMULATE KMC INSTR. LIST FOR.
1665 005046 004537 012100 JSR R5,KMSIM ;/COMMAND=NOP.
1666 000001 .MD.=1
1667 ;/COMMAND=NOP.
1668 005052 000001 .WORD .MD.
1669 ;/RETURN HERE AFTER COMMAND .
1670
1671 005054 012737 000004 001124 MOV #4,$GDDAT ;/RECORD XFERR PATTERN
1672 005062 113777 001124 174320 MOVB $GDDAT,@BSEL3 ;/WRITE PATTERN TO u CODE.
1673 005070 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1674 000007 .MD.=7
1675 ;/WRITE FAST PATH REG.
1676 005074 000007 .WORD .MD.
1677 ;/RETURN HERE AFTER COMMAND
1678 ;/MCMC-
1679 005076 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1680 005102 001424 .WORD KMCSR
1681 000006 .MD.=6
1682 ;/READ FAST PATH REG.
1683 005104 000006 .WORD .MD.
1684 ;/RETURN HERE AFTER COMMAND .
1685

```

1686 005106 117737 174326 001126  
1687 005114 023737 001124 001126  
1688 005122 001401

MOVB @BSEL3,\$BDDAT ;/READ SIDE #1  
CMP \$GDDAT,\$BDDAT ;/DATA SENT=DATA RECEIVED?  
BEQ 1\$

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1692 005124 104003  
1693

ERROR 3 ;INPROPER XFER OF PATTERN 4 FROM  
;/SIDE 2 TO SIDE 1

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1697 005126 005037 001124 1\$:  
1698 005132 105077 174252  
1699 005136 004537 012100  
1700 000007

CLR \$GDDAT ;/NOW XFER A ZERO PATTERN  
CLRB @MBSEL3  
JSR RS,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=7 ;/WRITE FAST PATH REG.

1701  
1702 005142 000007

.WORD .MD.  
;/RETURN HERE AFTER COMMAND

1703  
1704  
1705 005144 004537 011656  
1706 005150 001424  
1707 000006

JSR RS,ISSUEC ;/ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=6

1708  
1709 005152 000006

;/READ FAST PATH REG.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND .

1710  
1711  
1712 005154 117737 174260 001126  
1713 005162 001401

MOVB @BSEL3,\$BDDAT ;/READ PATTERN ERROR IF NON-ZERO  
BEQ TST24 ;;

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1717 005164 104003

ERROR 3 ;FAILED TO XFER ZERO PATTERN.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1721  
1722  
1723 ;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
1724 ;\*TEST 24 \*TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH RE  
1725 ;\*  
1726 ;\*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2  
1727 ;\*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A  
1728 ;\*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.  
1729 ;\*

1730 005166 000004  
1731  
1732 005170 004537 012100  
1733 000001

1\$TST24: SCOPE ;/DXFRF-  
JSR RS,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=1 ;/COMMAND=NOP.

1734  
1735 005174 000001

.WORD .MD.  
;/RETURN HERE AFTER COMMAND .

1736  
1737  
1738 005176 012737 000010 001124  
1739 005204 113777 001124 174176

MOV #10,\$GDDAT ;/RECORD XFERR PATTERN  
MOVB \$GDDAT,@MBSEL3 ;/WRITE PATTERN TO U CODE.

H04

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 34  
T24 \*TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0046

```

1740 005212 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1741 000007 .MD.=7 ;/WRITE FAST PATH REG.
1742 ;/RETURN HERE AFTER COMMAND
1743 005216 000007 .WORD .MD. ;/MCMC-
1744 ;/RETURN HERE AFTER COMMAND ;/ISSUE COMMAND TO KMC #1.
1745 JSR R5,ISSUEC ;/MCMC-
1746 005220 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1747 005224 001424 .WORD KMCSR ;/READ FAST PATH REG.
1748 000006 .MD.=6 ;/READ FAST PATH REG.
1749 ;/RETURN HERE AFTER COMMAND .
1750 005226 000006 .WORD .MD. ;/RETURN HERE AFTER COMMAND .
1751 ;/RETURN HERE AFTER COMMAND .
1752
1753 005230 117737 174204 001126 MOVB @BSEL3,$BDDAT ;/READ SIDE #1
1754 005236 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
1755 005244 001401 BEQ IS

```

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1759 005246 104003 ERROR 3 ;INPROPER XFER OF PATTERN 10 FROM
1760 ;SIDE 2 TO SIDE 1

```

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1764 005250 005037 001124 1S: CLR $GDDAT ;/NOW XFER A ZERO PATTERN
1765 005254 105077 174130 CLR @BSEL3
1766 005260 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
1767 000007 .MD.=7 ;/WRITE FAST PATH REG.
1768 ;/RETURN HERE AFTER COMMAND
1769 005264 000007 .WORD .MD. ;/MCMC-
1770 ;/RETURN HERE AFTER COMMAND ;/ISSUE COMMAND TO KMC #1.
1771 JSR R5,ISSUEC ;/MCMC-
1772 005266 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
1773 005272 001424 .WORD KMCSR ;/READ FAST PATH REG.
1774 000006 .MD.=6 ;/READ FAST PATH REG.
1775 ;/RETURN HERE AFTER COMMAND .
1776 005274 000006 .WORD .MD. ;/RETURN HERE AFTER COMMAND .
1777 ;/RETURN HERE AFTER COMMAND .
1778
1779 005276 117737 174136 001126 MOVB @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1780 005304 001401 BEQ TST25 ;

```

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1784 005306 104003 ERROR 3 ;FAILED TO XFER ZERO PATTERN.

```

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

1788
1789 ;*****
1790 ;*TEST 25 *TEST THAT PATTERN 20 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH RE
1791 ;*
1792 ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
1793 ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A

```

```

1794
1795
1796
1797 005310 000004
1798
1799 005312 004537 012100
1800 000001
1801
1802 005316 000001
1803
1804
1805 005320 012737 000020 001124
1806 005326 113777 001124 174054
1807 005334 004537 012100
1808 000007
1809
1810 005340 000007
1811
1812
1813 005342 004537 011656
1814 005346 001424
1815 000006
1816
1817 005350 000006
1818
1819
1820 005352 117737 174062 001126
1821 005360 023737 001124 001126
1822 005366 001401

1826 005370 104003
1827

1831 005372 005037 001124
1832 005376 105077 174006
1833 005402 004537 012100
1834 000007
1835
1836 005406 000007
1837
1838
1839 005410 004537 011656
1840 005414 001424
1841 000006
1842
1843 005416 000006
1844
1845
1846 005420 117737 174014 001126
1847 005426 001401

```

```

;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
;*
*****
†ST25: SCOPE
        JSR      R5,KMSIM          ;/-DXFRF-
        .MD.=1                    ;/DMC SIMULATE KMC INSTR. LIST FOR.
                                   ;/COMMAND=NOP.
        .WORD   .MD.
        ;/RETURN HERE AFTER COMMAND .
        MOV     #20,$GDDAT         ;/RECORD XFERR PATTERN
        MOVB   $GDDAT,@MBSEL3     ;/WRITE PATTERN TO U CODE.
        JSR    R5,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
        .MD.=7
                                   ;/WRITE FAST PATH REG.
        .WORD   .MD.
        ;/RETURN HERE AFTER COMMAND .
        JSR    R5,ISSUEC         ;/-MCMD-
        .WORD   KMCSR           ;/ISSUE COMMAND TO KMC #1.
        .MD.=6
                                   ;/READ FAST PATH REG.
        .WORD   .MD.
        ;/RETURN HERE AFTER COMMAND .
        MOVB   @MBSEL3,$BDDAT     ;/READ SIDE #1
        CMP    $GDDAT,$BDDAT     ;/DATA SENT=DATA RECEIVED?
        BEQ    1$

;*****>> ERROR <<*****
ERROR 3                               ;INPROPER XFER OF PATTERN 20 FROM
                                   ;/SIDE 2 TO SIDE 1

;*****>> ERROR <<*****
1$: CLR     $GDDAT                ;/NOW XFER A ZERO PATTERN
   CLRB   @MBSEL3
   JSR    R5,KMSIM              ;/DMC SIMULATE KMC INSTR. LIST FOR.
   .MD.=7
                                   ;/WRITE FAST PATH REG.
   .WORD   .MD.
   ;/RETURN HERE AFTER COMMAND .
   JSR    R5,ISSUEC            ;/-MCMD-
   .WORD   KMCSR              ;/ISSUE COMMAND TO KMC #1.
   .MD.=6
                                   ;/READ FAST PATH REG.
   .WORD   .MD.
   ;/RETURN HERE AFTER COMMAND .
   MOVB   @MBSEL3,$BDDAT       ;/READ PATTERN ERROR IF NON-ZERO
   BEQ    TST26                ;;

```



1851 005430 104003

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
ERROR 3 ;FAILED TO XFER ZERO PATTERN.  
;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862

::\*\*\*\*\*  
;\*TEST 26 \*TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH RE  
;\*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2  
;\*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A  
;\*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.  
;\*\*\*\*\*

1863  
1864 005432 000004

†ST26: SCOPE  
JSR R5,KMSIM ;/-DXFRF-  
.MD.=1 ;/DMC SIMULATE KMC INSTR. LIST FOR.  
; /COMMAND=NOP.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND .

1865  
1866 005434 004537 012100  
1867 000001  
1868  
1869 005440 000001  
1870  
1871  
1872 005442 012737 000040 001124  
1873 005450 113777 001124 173732  
1874 005456 004537 012100  
1875 000007

MOV #40,\$GDDAT ;/RECORD XFERR PATTERN  
MOVB \$GDDAT,2MBSEL3 ;/WRITE PATTERN TO U CODE.  
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=7 ;/WRITE FAST PATH REG.

1876  
1877 005462 000007  
1878  
1879  
1880 005464 004537 011656  
1881 005470 001424  
1882 000006

.WORD .MD.  
;/RETURN HERE AFTER COMMAND .  
JSR R5,ISSUEC ;/-MCM-  
.WORD KMCSR ;/ISSUE COMMAND TO KMC #1.  
.MD.=6 ;/READ FAST PATH REG.

1883  
1884 005472 000006  
1885  
1886  
1887 005474 117737 173740 001126  
1888 005502 023737 001124 001126  
1889 005510 001401

.WORD .MD.  
;/RETURN HERE AFTER COMMAND .  
MOVB 2MBSEL3,\$BDDAT ;/READ SIDE #1  
CMP \$GDDAT,\$BDDAT ;/DATA SENT=DATA RECEIVED?  
BEQ 1\$

1893 005512 104003  
1894

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
ERROR 3 ;INPROPER XFER OF PATTERN 40 FROM  
;/SIDE 2 TO SIDE 1  
;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1898 005514 005037 001124  
1899 005520 105077 173664  
1900 005524 004537 012100  
1901 000007

1\$: CLR \$GDDAT ;/NOW XFER A ZERO PATTERN  
CLRB 2MBSEL3  
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=7

# K04

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 37  
T26 \*TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0049

```

1902                                     ;/WRITE FAST PATH REG.
1903 005530 000007                       .WORD .MD.
1904                                     ;/RETURN HERE AFTER COMMAND .
1905                                     ;/-MCMD-
1906 005532 004537 011656                 JSR    R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1907 005536 001424                       .WORD  KMCSR
1908 000006                               .MD.=6
1909                                     ;/READ FAST PATH REG.
1910 005540 000006                       .WORD .MD.
1911                                     ;/RETURN HERE AFTER COMMAND .
1912
1913 005542 117737 173672 001126         MOVB   @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1914 005550 001401                       BEQ    TST27         ;;
                                     ;*****>> ERROR <<*****
1918 005552 104003                       ERROR  3             ;FAILED TO xFER ZERO PATTERN.
                                     ;*****>> ERROR <<*****

1922                                     ;*****
1923                                     ;*TEST 27 *TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH R
1924                                     ;*
1925                                     ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
1926                                     ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
1927                                     ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1928                                     ;*
1929                                     ;*****
1930                                     ;*****
1931 005554 000004                         †TST27: SCOPE
1932                                     ;/-DXFRF-
1933 005556 004537 012100                 JSR    R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
1934 000001                               .MD.=1
1935                                     ;/COMMAND=NOP.
1936 005562 000001                       .WORD .MD.
1937                                     ;/RETURN HERE AFTER COMMAND .
1938
1939 005564 012737 000100 001124         MOV    #100,$GDDAT   ;/RECORD XFERR PATTERN
1940 005572 113777 001124 173610         MOVB   $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
1941 005600 004537 012100                 JSR    R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
1942 000007                               .MD.=7
1943                                     ;/WRITE FAST PATH REG.
1944 005604 000007                       .WORD .MD.
1945                                     ;/RETURN HERE AFTER COMMAND .
1946                                     ;/-MCMD-
1947 005606 004537 011656                 JSR    R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1948 005612 001424                       .WORD  KMCSR
1949 000006                               .MD.=6
1950                                     ;/READ FAST PATH REG.
1951 005614 000006                       .WORD .MD.
1952                                     ;/RETURN HERE AFTER COMMAND .
1953
1954 005616 117737 173616 001126         MOVB   @BSEL3,$BDDAT ;/READ SIDE #1
1955 005624 023737 001124 001126         CMP    $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?

```

L04

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 38  
T27 \*TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0050

1956 005632 001401

BEQ 15  
;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1960 005634 104003  
1961

ERROR 3 ;INPROPER XFER OF PATTERN 100 FROM  
; /SIDE 2 TO SIDE 1

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1965 005636 005037 001124  
1966 005642 105077 173542  
1967 005646 004537 012100  
1968 000007  
1969

15: CLR \$GDDAT ; /NOW XFER A ZERO PATTERN  
CLR @MBSEL3 ; /DMC SIMULATE KMC INSTR. LIST FOR.  
JSR R5, KMSIM ; /WRITE FAST PATH REG.  
.MD.=7

1970 005652 000007

.WORD .MD.  
; /RETURN HERE AFTER COMMAND

1971  
1972  
1973 005654 004537 011656  
1974 005660 001424  
1975 000006  
1976

JSR R5, ISSUEC ; /-MCMD-  
; /ISSUE COMMAND TO KMC #1.  
.WORD KMCSR

1977 005662 000006

.WORD .MD.  
; /READ FAST PATH REG.

1978

.WORD .MD.  
; /RETURN HERE AFTER COMMAND .

1979  
1980 005664 117737 173550 001126  
1981 005672 001401

MOV @MBSEL3, \$BDDAT ; /READ PATTERN ERROR IF NON-ZERO  
BEQ TST30 ; ;

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1985 005674 104003

ERROR 3 ; FAILED TO XFER ZERO PATTERN.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

1989  
1990  
1991 ;\*\*\*\*\*>> \*TEST 30 \*TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH R  
1992 ;\*  
1993 ;\*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2  
1994 ;\*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A  
1995 ;\*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.  
1996 ;\*  
1997 ;\*\*\*\*\*>>

1998 005676 000004  
1999  
2000 005700 004537 012100  
2001 000001  
2002

TST30: SCOPE ; /-DXFRF-  
JSR R5, KMSIM ; /DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=1 ; /COMMAND=NOP.

2003 005704 000001

.WORD .MD.  
; /RETURN HERE AFTER COMMAND .

2004  
2005  
2006 005706 012737 000200 001124  
2007 005714 113777 001124 173466  
2008 005722 004537 012100  
2009 000007

MOV #200, \$GDDAT ; /RECORD XFERR PATTERN  
MOV \$GDDAT, @MBSEL3 ; /WRITE PATTERN TO u CODE.  
JSR R5, KMSIM ; /DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=7

M04

MD-11-DRLPN-A MACY11 27(654) 15-DEC-77 08:43 PAGE 39  
DRLPN.P11 T30 \*TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG

SEQ 0051

```

2010                                     ;/WRITE FAST PATH REG.
2011 005726 000007 .WORD .MD.
2012                                     ;/RETURN HERE AFTER COMMAND
2013                                     ;/-MCMD-
2014 005730 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2015 005734 001424 .WORD KMCSR
2016 000006 .MD.=6
2017                                     ;/READ FAST PATH REG.
2018 005736 000006 .WORD .MD.
2019                                     ;/RETURN HERE AFTER COMMAND .
2020
2021 005740 117737 173474 001126 MOVB @BSEL3,$BDDAT ;/READ SIDE #1
2022 005746 023737 001124 001126 CMP $GDDAT,$BDDAT ;/DATA SENT=DATA RECEIVED?
2023 005754 001401 BEQ IS

;*****>> ERROR <<*****
2027 005756 104003 ERROR 3 ;INPROPER XFER OF PATTERN 200 FROM
2028                                     ;/SIDE 2 TO SIDE 1

;*****>> ERROR <<*****
2032 005760 005037 001124 IS: CLR $GDDAT ;/NOW XFER A ZERO PATTERN
2033 005764 105077 173420 CLRB @BSEL3
2034 005770 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2035 000007 .MD.=7
2036                                     ;/WRITE FAST PATH REG.
2037 005774 000007 .WORD .MD.
2038                                     ;/RETURN HERE AFTER COMMAND
2039                                     ;/-MCMD-
2040 005776 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2041 006002 001424 .WORD KMCSR
2042 000006 .MD.=6
2043                                     ;/READ FAST PATH REG.
2044 006004 000006 .WORD .MD.
2045                                     ;/RETURN HERE AFTER COMMAND .
2046
2047 006006 117737 173426 001126 MOVB @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
2048 006014 001401 BEQ TST31

;*****>> ERROR <<*****
2052 006016 104003 ERROR 3 ;FAILED TO XFER ZERO PATTERN.

;*****>> ERROR <<*****
2056
2057 ;*****
2058 ;*TEST 31 *TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA SILO
2059 ;*****
2060 006020 000004 TST31: SCOPE
2061
2062 006022 004737 012524 JSR PC,BINT ;INIT. KMCS.
2063 006026 112777 000377 173404 MOVB #377,@BSEL3 ;DATA TO BE XFERRED.

```

N04

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 40  
T31 \*TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA SILO

SEQ 0052

```

2064
2065 006034 004537 011656 JSR R5,ISSUEC ;/-MCMD-
2066 006040 001424 .WORD KMCSR ;/ISSUE COMMAND TO KMC #1.
2067 000005 .MD.=5
2068 ;/WRITE TO SILO.
2069 006042 000005 .WORD .MD.
2070 ;/RETURN HERE AFTER COMMAND
2071 006044 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2072 000001 .MD.=1
2073 ;/COMMAND=NOP.
2074 006050 000001 .WORD .MD.
2075 ;/RETURN HERE AFTER COMMAND
2076 006052 117737 173336 001126 MOVB #MBSEL5,$BDDAT ;READ SIDE 2 IOSR
2077 006060 012737 000054 001124 MOV #54,$GDDAT ;EXPECT BIT1 TO BE CLEAR.
2078 006066 023737 001126 001124 CMP $BDDAT,$GDDAT ;DID BIT1 CLEAR?
2079 006074 001402 BEQ 1$ ;YES - NEXT CHECK

;*****>> ERROR <<*****

2083 006076 104002 ERROR 2 ;BIT1 IOSR SIDE 2 FAILED
2084 ;TO CLEAR (INDICATING SILO DATA
2085 ;PRESENT) WHEN DATA WAS
2086 ;WRITTEN INTO SIDE 1 SILO.

;*****>> ERROR <<*****

2090 006100 000413 BR TST32 ;;
2091
2092 006102 1$:
2093 006102 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2094 000004 .MD.=4 ;/READ SILO.
2095
2096 006106 000004 .WORD .MD.
2097 ;/RETURN HERE AFTER COMMAND
2098 006110 117737 173274 001126 MOVB #MBSEL3,$BDDAT ;READ DATA, IF NON-ZERO - NO ERROR.
2099 006116 001004 BNE TST32
2100 006120 012737 000377 001124 MOV #377,$GDDAT ;HAD EXPECTED 377 BUT WOULD
2101 ;HAVE SETTLED FOR ANYTHING TO
2102 ;INDICATE SOME DATA GOT THROUGH

;*****>> ERROR <<*****

2106 006126 104006 ERROR 6 ;FAILED TO XFER DATA FROM SIDE 1
2107 ;TO SIDE 2 VIA SILO (FIFO) REG.

;*****>> ERROR <<*****

2111
2112
2113 ;*****
2114 ;*TEST 32 *TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA SILO
2115 006130 000004 TST32: SCOPE
2116
2117 006132 004737 012524 JSR PC,BINT ;INIT KMCS.

```

```

2118 006136 112777 000377 173244      MOVB    #377, @BSEL3      ;DATA TO BE XFERRD.
2119 006144 004537 012100                JSR     R5, KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2120 000005                                .MD.=5
2121                                ;/WRITE TO SILO.
2122 006150 000005                                .WORD  .MD.
2123                                ;/RETURN HERE AFTER COMMAND
2124 006152 004537 012100                JSR     R5, KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2125 000001                                .MD.=1
2126                                ;/COMMAND=NOP.
2127 006156 000001                                .WORD  .MD.
2128                                ;/RETURN HERE AFTER COMMAND
2129 006160 117737 173260 001126          MOVB    @BSEL5, $BDDAT   ;READ SIDE 1 UBSR
2130 006166 012737 000050 001124          MOV     #50, $GDDAT     ;EXPECT BIT0 TO BE CLEAR.
2131 006174 023737 001124 001126          CMP     $GDDAT, $BDDAT  ;DID BIT0 CLEAR?
2132 006202 001402                                BEQ     1$

```

```

;*****>> ERROR <<*****
ERROR 1                                ;BIT 1 UBSR SIDE 1 FAILED
;TO CLEAR (INDICATION SILO DATA
;PRESENT) WHEN DATA WAS
;WRITTEN INTO SIDE 2 SILO

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2143 006206 000414                                BR      TST33            ;;
2144 006210                                1$:
2145                                ;/-MCM-
2146 006210 004537 011656                JSR     R5, ISSUEC      ;/ISSUE COMMAND TO KMC #1.
2147 006214 001424                                .WORD  KMCSR
2148 000004                                .MD.=4
2149                                ;/READ SILO.
2150 006216 000004                                .WORD  .MD.
2151                                ;/RETURN HERE AFTER COMMAND
2152 006220 117737 173214 001126          MOVB    @BSEL3, $BDDAT  ;READ DATA.
2153 006226 001004                                BNE    TST33            ;;
2154
2155 006230 012737 000377 001124          MOV     #377, $GDDAT    ;HAD EXPECTED 377 BUT WOULD
2156                                ;HAVE SETTLED FOR ANYTHING TO
2157                                ;INDICATE SOME DATA GOT THROUGH.

```

```

;*****>> ERROR <<*****
ERROR 5                                ;FAILED TO XFER DATA FROM SIDE 2
;TO SIDE 1 VIA SILO (FIFO) REG

```

```

;*****
;TEST 33 *TEST THAT DATA PATTERN 0 CAN BE XFERRD VIA SILO FROM SIDE 1 TO SIDE 2
;*****
TST33: SCOPE

```

```

2168 006240 000004                                ;/-SIDA-
2169                                ;/RECORD XFER 0
2170 006242 012737 000000 001124          MOV     #0, $GDDAT
2171 006250 113777 001124 173162          MOVB    $GDDAT, @BSEL3 ;/DATA TO XFERR.

```

C05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T33

27(654) 15-DEC-77 08:43 PAGE 42

\*TEST THAT DATA PATTERN 0 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE 2

SEQ 0054

```

2172
2173 006256 004537 011656 JSR R5,ISSUEC ;/-MCMD-
2174 006262 001424 .WORD KMCSR ;/ISSUE COMMAND TO KMC #1.
2175 000005 .MD.=5
2176 ;/WRITE TO SILO.
2177 006264 000005 .WORD .MD.
2178 ;/RETURN HERE AFTER COMMAND
2179 006266 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2180 000004 .MD.=4
2181 ;/READ SILO.
2182 006272 000004 .WORD .MD.
2183 ;/RETURN HERE AFTER COMMAND .
2184
2185 006274 117737 173110 001126 MOVB @MBSEL3,$BDDAT ;/READ FROM REG.
2186 006302 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2187 006310 001401 BEQ TST34 ;
2188

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2192 006312 104006 ERROR 6 ;/SIDE 1 TO SIDE 2 DATA
2193 ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2197
2198 ;*****
2199 ;*TEST 34 *TEST THAT DATA PATTERN 1 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE 2
2200 ;*****

```

```

2201 006314 000004 †TST34: SCOPE
2202
2203 006316 012737 000001 001124 MOV #1,$GDDAT ;/-SIDA-
2204 006324 113777 001124 173106 MOVB $GDDAT,@BSEL3 ;/RECORD XFER 1
2205 ;/DATA TO XFERR.
2206 006332 004537 011656 JSR R5,ISSUEC ;/-MCMD-
2207 006336 001424 .WORD KMCSR ;/ISSUE COMMAND TO kMC #1.
2208 000005 .MD.=5
2209 ;/WRITE TO SILO.
2210 006340 000005 .WORD .MD.
2211 ;/RETURN HERE AFTER COMMAND
2212 006342 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2213 000004 .MD.=4
2214 ;/READ SILO.
2215 006346 000004 .WORD .MD.
2216 ;/RETURN HERE AFTER COMMAND .
2217
2218 006350 117737 173034 001126 MOVB @MBSEL3,$BDDAT ;/READ FROM REG.
2219 006356 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2220 006364 001401 BEQ TST35 ;
2221

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2225 006366 104006 ERROR 6 ;/SIDE 1 TO SIDE 2 DATA

```

2226

;/XFERR ERROR VIA SILO.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254

006370 000004  
006372 012737 000002 001124  
006400 113777 001124 173032  
006406 004537 011656  
006412 001424 000005  
006414 000005  
006416 004537 012100 000004  
006422 000004  
006424 117737 172760 001126  
006432 123737 001124 001126  
006440 001401

```
*****  
*TEST 35 *TEST THAT DATA PATTERN 2 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2  
*****  
↑ST35: SCOPE  
MOV #2,$GDDAT ;/-SIDA-  
MOVSB $GDDAT,$BSEL3 ;/RECORD XFER 2  
; /DATA TO XFERR.  
JSR R5,ISSUEC ;/-MCMD-  
; /ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=5 ;/WRITE TO SILO.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND  
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=4 ;/READ SILO.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND .  
MOVSB $BSEL3,$BDDAT ;/READ FROM REG.  
CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?  
BEQ TST36 ;;
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2258 006442 104006  
2259

ERROR 6 ;/SIDE 1 TO SIDE 2 DATA  
;/XFERR ERROR VIA SILO.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279

006444 000004  
006446 012737 000004 001124  
006454 113777 001124 172756  
006462 004537 011656  
006466 001424 000005  
006470 000005  
006472 004537 012100 000004

```
*****  
*TEST 36 *TEST THAT DATA PATTERN 4 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2  
*****  
↑ST36: SCOPE  
MOV #4,$GDDAT ;/-SIDA-  
MOVSB $GDDAT,$BSEL3 ;/RECORD XFER 4  
; /DATA TO XFERR.  
JSR R5,ISSUEC ;/-MCMD-  
; /ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=5 ;/WRITE TO SILO.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND  
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=4
```



```

2280                                     ;/READ SILO.
2281 006476 000004 .WORD .MD.
2282                                     ;/RETURN HERE AFTER COMMAND .
2283
2284 006500 117737 172704 001126 MOVB 2MBSEL3,$BDDAT ;/READ FROM REG.
2285 006506 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2286 006514 001401 BEQ TST37 ;
2287

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2291 006516 104006 ERROR 6 ;/SIDE 1 TO SIDE 2 DATA
2292                                     ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2296
2297 ;:*****
2298 ;*TEST 37 *TEST THAT DATA PATTERN 10 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2299 ;:*****
2300 006520 000004 †TST37: SCOPE

```

```

2301                                     ;/-SIDA-
2302 006522 012737 000010 001124 MOV #10,$GDDAT ;/RECORD XFER 10
2303 006530 113777 001124 172702 MOVB $GDDAT,2MBSEL3 ;/DATA TO XFERR.
2304                                     ;/-MCMD-
2305 006536 004537 011656 JSR RS,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2306 006542 001424 .WORD KMCSR
2307 000005 .MD.=5
2308                                     ;/WRITE TO SILO.
2309 006544 000005 .WORD .MD.
2310                                     ;/RETURN HERE AFTER COMMAND
2311 006546 004537 012100 JSR RS,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2312 000004 .MD.=4
2313                                     ;/READ SILO.
2314 006552 000004 .WORD .MD.
2315                                     ;/RETURN HERE AFTER COMMAND .
2316
2317 006554 117737 172630 001126 MOVB 2MBSEL3,$BDDAT ;/READ FROM REG.
2318 006562 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2319 006570 001401 BEQ TST40 ;
2320

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2324 006572 104006 ERROR 6 ;/SIDE 1 TO SIDE 2 DATA
2325                                     ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2329
2330 ;:*****
2331 ;*TEST 40 *TEST THAT DATA PATTERN 20 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2332 ;:*****
2333 006574 000004 †TST40: SCOPE

```

# F05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T40

27(654) 15-DEC-77 08:43 PAGE 45  
\*TEST THAT DATA PATTERN 20 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE 2

SEQ 0057

```

2334
2335 006576 012737 000020 001124      MOV      #20,$GDDAT      ;/-SIDA-
2336 006604 113777 001124 172626      MOVVB   $GDDAT,$BSEL3 ;/RECORD XFER 20
2337                                     ;/DATA TO XFERR.
2338 006612 004537 011656      JSR      R5,ISSUEC      ;/-MCMD-
2339 006616 001424 000005      .WORD   KMCSR          ;/ISSUE COMMAND TO KMC #1.
2340                                     .MD.=5
2341                                     ;/WRITE TO SILO.
2342 006620 000005      .WORD   .MD.
2343 ;/RETURN HERE AFTER COMMAND
2344 006622 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
2345 000004      .MD.=4
2346                                     ;/READ SILO.
2347 006626 000004      .WORD   .MD.
2348 ;/RETURN HERE AFTER COMMAND .
2349
2350 006630 117737 172554 001126      MOVVB   $MBSEL3,$BDDAT ;/READ FROM REG.
2351 006636 123737 001124 001126      CMPB    $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2352 006644 001401      BEQ     TST41          ;;
2353

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2357 006646 104006      ERROR   6              ;/SIDE 1 TO $IDE 2 DATA
2358                                     ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2362
2363 ;*****
2364 ;*TEST 41 *TEST THAT DATA PATTERN 40 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO $IDE 2
2365 ;*****
2366 006650 000004      TST41: SCOPE
2367
2368 006652 012737 000040 001124      MOV      #40,$GDDAT      ;/-SIDA-
2369 006660 113777 001124 172552      MOVVB   $GDDAT,$BSEL3 ;/RECORD XFER 40
2370                                     ;/DATA TO XFERR.
2371 006666 004537 011656      JSR      R5,ISSUEC      ;/-MCMD-
2372 006672 001424 000005      .WORD   KMCSR          ;/ISSUE COMMAND TO KMC #1.
2373                                     .MD.=5
2374                                     ;/WRITE TO SILO.
2375 006674 000005      .WORD   .MD.
2376 ;/RETURN HERE AFTER COMMAND
2377 006676 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
2378 000004      .MD.=4
2379                                     ;/READ SILO.
2380 006702 000004      .WORD   .MD.
2381 ;/RETURN HERE AFTER COMMAND .
2382
2383 006704 117737 172500 001126      MOVVB   $MBSEL3,$BDDAT ;/READ FROM REG.
2384 006712 123737 001124 001126      CMPB    $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2385 006720 001401      BEQ     TST42          ;;
2386

```

G05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T41

27(654) 15-DEC-77 08:43 PAGE 46

\*TEST THAT DATA PATTERN 40 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE 2

SEQ 0058

2390 006722 104006  
2391

\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
ERROR 6 ;/SIDE 1 TO SIDE 2 DATA  
;/XFERR ERROR VIA SILO.

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419

006724 000004  
006726 012737 000100 001124  
006734 113777 001124 172476  
006742 004537 011656  
006746 001424  
000005  
006750 000005  
006752 004537 012100  
000004  
006756 000004  
000004  
006760 117737 172424 001126  
006766 123737 001124 001126  
006774 001401

\*\*\*\*\*  
\*TEST 42 \*TEST THAT DATA PATTERN 100 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE  
\*\*\*\*\*

TST42: SCOPE  
MOV #100,\$GDDAT ;/-SIDA-  
MOV B \$GDDAT, @BSEL3 ;/RECORD XFER 100  
; DATA TO XFERR.  
JSR R5, ISSUCC ;/-MCMD-  
; /ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=5 ;/WRITE TO SILO.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND  
JSR R5, KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=4 ;/READ SILO.  
.WORD .MD.  
;/RETURN HERE AFTER COMMAND .  
MOV B @BSEL3, \$BDDAT ;/READ FROM REG.  
CMP B \$GDDAT, \$BDDAT ;/DATA WRITTEN = DATA RECEIVED?  
BEQ TST43 ;;

\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
ERROR 6 ;/SIDE 1 TO SIDE 2 DATA  
;/XFERR ERROR VIA SILO.

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2428  
2429  
2430  
2431  
2432  
2433  
2434  
2435  
2436  
2437  
2438  
2439  
2440  
2441

007000 000004  
007002 012737 000200 001124  
007010 113777 001124 172422  
007016 004537 011656  
007022 001424  
000005  
007024 000005

\*\*\*\*\*  
\*TEST 43 \*TEST THAT DATA PATTERN 200 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE  
\*\*\*\*\*

TST43: SCOPE  
MOV #200,\$GDDAT ;/-SIDA-  
MOV B \$GDDAT, @BSEL3 ;/RECORD XFER 200  
; DATA TO XFERR.  
JSR R5, ISSUCC ;/-MCMD-  
; /ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=5 ;/WRITE TO SILO.  
.WORD .MD.

# H05

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 47  
T43 \*TEST THAT DATA PATTERN 200 CAN BE XFERRERD VIA SILO FROM SIDE 1 TO SIDE 2

SEQ 0059

```
2442      ;/RETURN HERE AFTER COMMAND .
2443 007026 004537 012100 JSR    RS,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
2444      .MD.=4
2445      ;/READ SILO.
2446 007032 000004
2447      .WORD .MD.
2448      ;/RETURN HERE AFTER COMMAND .
2449 007034 117737 172350 001126 MOVB   @MSEL3,$BDDAT ;/READ FROM REG.
2450 007042 123737 001124 001126 CMPB   $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2451 007050 001401
2452      BEQ    TST44      ;;
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
2456 007052 104006      ERROR    6      ;/SIDE 1 TO SIDE 2 DATA
2457      ;/XFERR ERROR VIA SILO.
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
2461
2462
2463
2464      ;*****
2465      ;*TEST 44      *TEST THAT DATA PATTERN 0 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1
2466      ;*****
2467      TST44: SCOPE
```

```
2468 007054 000004
2469 007056 012737 000000 001124 MOV    #0,$GDDAT      ;/-SIDA-
2470 007064 113777 001124 172316 MOVB   $GDDAT,@MSEL3 ;/RECORD XFER 0
2471 007072 004537 012100 JSR    RS,KMSIM      ;/DATA TO XFERR.
2472      .MD.=5      ;/DMC SIMULATE KMC INSTR. LIST FOR.
2473      ;/WRITE TO SILO.
2474      .WORD .MD.
2475      ;/RETURN HERE AFTER COMMAND .
2476 007100 004537 011656 JSR    RS,ISSUEC     ;/-MCMD-
2477 007104 001424 ;/ISSUE COMMAND TO KMC #1.
2478      .WORD KMC5R
2479      .MD.=4
2480 007106 000004      ;/READ SILO.
2481      .WORD .MD.
2482      ;/RETURN HERE AFTER COMMAND .
2483 007110 117737 172324 001126 MOVB   @MSEL3,$BDDAT ;/READ FROM REG.
2484 007116 123737 001124 001126 CMPB   $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2485 007124 001401      BEQ    TST45      ;;
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
2489 007126 104005      ERROR    5      ;/SIDE 2 TO SIDE 1 DATA
2490      ;/XFERR ERROR VIA SILO.
```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```
2494
2495      ;*****
```

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T45

27(654) 15-DEC-77 08:43 PAGE 48  
\*TEST THAT DATA PATTERN 1 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1

SEQ 0060

```

2496 ;*TEST 45 *TEST THAT DATA PATTERN 1 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1
2497 ;*****
2498 †T45: SCOPE
2499
2500 007130 000004 MOV #1,$GDDAT ;/-SIDA-
2501 007140 012737 000001 001124 MOVB $GDDAT,@MBSEL3 ;/RECORD XFER 1
2502 007146 113777 001124 172242 JSR R5,KMSIM ;/DATA TO XFERR.
2503 000005 .MD.=5 ;/DMC SIMULATE KMC INSTR. LIST FOR.
2504 ;/WRITE TO SILO.
2505 007152 000005 .WORD .MD.
2506 ;/RETURN HERE AFTER COMMAND
2507 ;/-MCMD-
2508 007154 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2509 007160 001424 .WORD KMCSR
2510 000004 .MD.=4
2511 ;/READ SILO.
2512 007162 000004 .WORD .MD.
2513 ;/RETURN HERE AFTER COMMAND .
2514
2515 007164 117737 172250 001126 MOVB @BSEL3,$BDDAT ;/READ FROM REG.
2516 007172 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2517 007200 001401 BEQ TST46 ;;
;*****>> ERROR <<*****
2521 007202 104005 ERROR 5 ;/SIDE 2 TO SIDE 1 DATA
2522 ;/XFERR ERROR VIA SILO.
;*****>> ERROR <<*****

2526 ;*****
2527 ;*TEST 46 *TEST THAT DATA PATTERN 2 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1
2528 ;*****
2529 †T46: SCOPE
2530
2531 007204 000004 MOV #2,$GDDAT ;/-SIDA-
2532 007206 012737 000002 001124 MOVB $GDDAT,@MBSEL3 ;/RECORD XFER 2
2533 007214 113777 001124 172166 JSR R5,KMSIM ;/DATA TO XFERR.
2534 007222 004537 012100 .MD.=5 ;/DMC SIMULATE KMC INSTR. LIST FOR.
2535 000005 ;/WRITE TO SILO.
2536
2537 007226 000005 .WORD .MD.
2538 ;/RETURN HERE AFTER COMMAND
2539 ;/-MCMD-
2540 007230 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2541 007234 001424 .WORD KMCSR
2542 000004 .MD.=4
2543 ;/READ SILO.
2544 007236 000004 .WORD .MD.
2545 ;/RETURN HERE AFTER COMMAND .
2546
2547 007240 117737 172174 001126 MOVB @BSEL3,$BDDAT ;/READ FROM REG.
2548 007246 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2549 007254 001401 BEQ TST47 ;;

```

J05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T46

27(654) 15-DEC-77 08:43 PAGE 49  
\*TEST THAT DATA PATTERN 2 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1

SEQ 0061

```

;*****>> ERROR <<*****
2553 007256 104005 ERROR 5 ;/SIDE 2 TO SIDE 1 DATA
2554 ;/XFERR ERROR VIA SILO.
;*****>> ERROR <<*****

2558
2559 ;*****
2560 ;*TEST 47 *TEST THAT DATA PATTERN 4 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1
2561 ;*****
2562 TST47: SCOPE
2563 ;/SIDA-
2564 007262 000004 MOV #4,$GDDAT ;/RECORD XFER 4
2565 007270 012737 000004 001124 MOVB $GDDAT,@MBSEL3 ;/DATA TO XFERR.
2566 007276 113777 001124 172112 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2567 004537 012100 .MD.=5 ;/WRITE TO SILO.
2568 ;/RETURN HERE AFTER COMMAND
2569 007302 000005 .WORD .MD.
2570 ;/MCM-
2571 ;/ISSUE COMMAND TO KMC #1.
2572 007304 004537 011656 JSR R5,ISSUEC
2573 007310 001424 .WORD KMCSR
2574 000004 .MD.=4 ;/READ SILO.
2575 ;/RETURN HERE AFTER COMMAND .
2576 007312 000004 .WORD .MD.
2577 ;/READ FROM REG.
2578 ;/DATA WRITTEN = DATA RECEIVED?
2579 007314 117737 172120 001126 MOVB @MBSEL3,$BDDAT
2580 007322 123737 001124 001126 CMPB $GDDAT,$BDDAT
2581 007330 001401 BEQ TST50

;*****>> ERROR <<*****
2585 007332 104005 ERROR 5 ;/SIDE 2 TO SIDE 1 DATA
2586 ;/XFERR ERROR VIA SILO.
;*****>> ERROR <<*****

2590
2591 ;*****
2592 ;*TEST 50 *TEST THAT DATA PATTERN 10 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1
2593 ;*****
2594 TST50: SCOPE
2595 ;/SIDA-
2596 007336 012737 000010 001124 MOV #10,$GDDAT ;/RECORD XFER 10
2597 007344 113777 001124 172036 MOVB $GDDAT,@MBSEL3 ;/DATA TO XFERR.
2598 007352 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2599 000005 .MD.=5 ;/WRITE TO SILO.
2600 ;/RETURN HERE AFTER COMMAND
2601 007356 000005 .WORD .MD.
2602 ;/MCM-
2603 ;/ISSUE COMMAND TO KMC #1.

```

K05

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 50  
TSO \*TEST THAT DATA PATTERN 10 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1

SEQ 0062

```

2604 007360 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2605 007364 001424 .WORD KMCSR
2606 000004 .MD.=4
2607 ;/READ SILO.
2608 007366 000004 .WORD .MD.
2609 ;/RETURN HERE AFTER COMMAND .
2610
2611 007370 117737 172044 001126 MOVB @BSEL3,$BDDAT ;/READ FROM REG.
2612 007376 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2613 007404 001401 BEQ TST51 ;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2617 007406 104005 ERROR 5 ;/SIDE 2 TO SIDE 1 DATA
2618 ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2622
2623
2624 :*****
2625 :*TEST 51 *TEST THAT DATA PATTERN 20 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2626 :*****
2627 †TST51: SCOPE

```

```

2628 007410 000004 ;/-SIDA-
2629 007412 012737 000020 001124 MOV #20,$GDDAT ;/RECORD XFER 20
2630 007420 113777 001124 171762 MOVB $GDDAT,@BSEL3 ;/DATA TO XFERR.
2631 007426 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2632 000005 .MD.=5
2633 ;/WRITE TO SILO.

```

```

2634 007432 000005 .WORD .MD.
2635 ;/RETURN HERE AFTER COMMAND .
2636 007434 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.
2637 007440 001424 .WORD KMCSR
2638 000004 .MD.=4
2639 ;/READ SILO.
2640 007442 000004 .WORD .MD.
2641 ;/RETURN HERE AFTER COMMAND .
2642

```

```

2643 007444 117737 171770 001126 MOVB @BSEL3,$BDDAT ;/READ FROM REG.
2644 007452 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2645 007460 001401 BEQ TST52 ;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2649 007462 104005 ERROR 5 ;/SIDE 2 TO SIDE 1 DATA
2650 ;/XFERR ERROR VIA SILO.

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2654 :*****
2655 :*TEST 52 *TEST THAT DATA PATTERN 40 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2656 :*****
2657

```

L05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T52

27(654) 15-DEC-77 08:43 PAGE 51  
\*TEST THAT DATA PATTERN 40 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE 1

SEQ 0063

```

2658 007464 000004          TST52: SCOPE
2659
2660 007466 012737 000040 001124      MOV      #40,$GDDAT      ;/-SIDA-
2661 007474 113777 001124 171706      MOVB    $GDDAT,MBSEL3 ;/RECORD XFER 40
2662 007502 004537 012100                JSR     R5,KMSIM      ;/DATA TO XFERR.
2663 000005                                .MD.=5                ;/DMC SIMULATE KMC INSTR. LIST FOR.
2664
2665 007506 000005                                .WORD   .MD.          ;/WRITE TO SILO.
2666                                ;/RETURN HERE AFTER COMMAND .
2667
2668 007510 004537 011656      JSR     R5,ISSUEC     ;/-MCMD-
2669 007514 001424                .WORD   KMCSR        ;/ISSUE COMMAND TO kMC #1.
2670 000004                                .MD.=4
2671
2672 007516 000004                                .WORD   .MD.          ;/READ SILO.
2673                                ;/RETURN HERE AFTER COMMAND .
2674
2675 007520 117737 171714 001126      MOVB    MBSEL3,$BDDAT ;/READ FROM REG.
2676 007526 123737 001124 001126      CMPB    $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2677 007534 001401                BEQ     TST53         ;;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2681 007536 104005          ERROR 5                ;/SIDE 2 TO SIDE 1 DATA
2682

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

2686
2687
2688
2689
2690 007540 000004          ;*****
2691                                ;/TEST 53          *TEST THAT DATA PATTERN 100 CAN BE XFERRERD VIA SILO FROM SIDE 2 TO SIDE
2692                                ;*****
2693                                †TST53: SCOPE

```

```

2692 007542 012737 000100 001124      MOV      #100,$GDDAT   ;/-SIDA-
2693 007550 113777 001124 171632      MOVB    $GDDAT,MBSEL3 ;/RECORD XFER 100
2694 007556 004537 012100                JSR     R5,KMSIM      ;/DATA TO XFERR.
2695 000005                                .MD.=5                ;/DMC SIMULATE KMC INSTR. LIST FOR.
2696
2697 007562 000005                                .WORD   .MD.          ;/WRITE TO SILO.
2698                                ;/RETURN HERE AFTER COMMAND .
2699
2700 007564 004537 011656      JSR     R5,ISSUEC     ;/-MCMD-
2701 007570 001424                .WORD   KMCSR        ;/ISSUE COMMAND TO KMC #1.
2702 000004                                .MD.=4
2703
2704 007572 000004                                .WORD   .MD.          ;/READ SILO.
2705                                ;/RETURN HERE AFTER COMMAND .
2706
2707 007574 117737 171640 001126      MOVB    MBSEL3,$BDDAT ;/READ FROM REG.
2708 007602 123737 001124 001126      CMPB    $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
2709 007610 001401                BEQ     TST54         ;;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*



M05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T53

27(654) 15-DEC-77 08:43 PAGE 52

\*TEST THAT DATA PATTERN 100 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1

SEQ 0064

2713 007612 104005  
2714

ERROR 5 ;/SIDE 2 TO SIDE 1 DATA  
;/XFERR ERROR VIA SILO.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741

007614 000004  
007616 012737 000200 001124  
007624 113777 001124 171556  
007632 004537 012100  
000005  
007636 000005  
007640 004537 011656  
007644 001424 000004  
007646 000004  
007650 117737 171564 001126  
007656 123737 001124 001126  
007664 001401

```

;*****
;TEST 54 *TEST THAT DATA PATTERN 200 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE
;*****
TST54: SCOPE
MOV #200,$GDDAT ;/-SIDA-
MOV #200,$GDDAT ;/RECORD XFER 200
MOV #200,$GDDAT ;/DATA TO XFERR.
JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
.MD.=5 ;/WRITE TO SILO.
.WORD .MD.
;/RETURN HERE AFTER COMMAND
JSR R5,ISSUEC ;/-MCMD-
.WORD KMCSR ;/ISSUE COMMAND TO KMC #1.
.MD.=4 ;/READ SILO.
.WORD .MD.
;/RETURN HERE AFTER COMMAND .
MOV #200,$GDDAT ;/READ FROM REG.
CMPB $GDDAT,$BDDAT ;/DATA WRITTEN = DATA RECEIVED?
BEQ TST55 ;

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2745 007666 104005  
2746

ERROR 5 ;/SIDE 2 TO SIDE 1 DATA  
;/XFERR ERROR VIA SILO.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760  
2761  
2762  
2763  
2764  
2765

007670 000004  
007672 012737 000010 001160  
007700 012700 000067  
007704  
007704 004537 011656  
007710 001424

```

;*****
;TEST 55 *TEST THAT 7/8 FLAG SETS IN SIDE 1 WHEN 56 WORDS WRITTEN IN SILO
;*****
TST55: SCOPE
MOV #10,$TIMES ;/DO 10 ITERATIONS
;/-S78F-
MOV #55.,R0 ;/SET TO DO 56 TIMES
1$:
JSR R5,ISSUEC ;/-MCMD-
.WORD KMCSR ;/ISSUE COMMAND TO KMC #1.

```

# N05

MD-11-DRLPN-A  
DRLPN.P11

MACY11  
T55

27(654) 15-DEC-77 08:43 PAGE 53  
\*TEST THAT 7/8 FLAG SETS IN SIDE 1 WHEN 56 WORDS WRITTEN IN SILO

SEQ 0065

```

2766      000001      .MD.=1
2767      ;/COMMAND=NOP.
2768 007712 000001      .WORD .MD.
2769      ;/RETURN HERE AFTER COMMAND
2770 007714 132777 000001 171522 BITB #BIT0, @BSELS ;/FIFO OUTPUT READY?
2771 007722 001370      BNE 1$
2772      ;/-MCMD-
2773 007724 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
2774 007730 001424      .WORD KMCSR
2775      .MD.=5
2776      ;/WRITE TO SILO.
2777 007732 000005      .WORD .MD.
2778      ;/RETURN HERE AFTER COMMAND .
2779 007734 005300      DEC R0
2780 007736 001362      BNE 1$ ;/DONE 55 TIMES?
2781
2782 007740 012737 000050 001124 MOV #50, $GDDAT ;/EXP'D IOSR
2783      ;/-MCMD-
2784 007746 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
2785 007752 001424      .WORD KMCSR
2786      .MD.=1
2787      ;/COMMAND=NOP.
2788 007754 000001      .WORD .MD.
2789      ;/RETURN HERE AFTER COMMAND
2790 007756 117737 171462 001126 MOVB @BSELS, $BDDAT ;/READ SR.
2791 007764 132777 000200 171452 BITB #BIT7, @BSELS ;/DID 7/8 FLAG SET?
2792 007772 001404      BEQ 2$ ;/NO - GOOD NEED 1 MORE CHARACTER

;*****>> ERROR <<*****

2796 007774 104001      ERROR 1 ;/ERROR 7/8 FLAG SET IN SIDE 1
2797      ;/SET WHEN ONLY 55 WORDS WRITTEN

;*****>> ERROR <<*****

2801 007776 004737 012524 JSR PC, BINT
2802 010002 000452      BR TST56 ;;
2803
2804 010004      2$:
2805
2806 010004 004537 011656 JSR R5, ISSUEC ;/-MCMD-
2807 010010 001424      .WORD KMCSR ;/ISSUE COMMAND TO KMC #1.
2808      .MD.=5
2809      ;/WRITE TO SILO.
2810 010012 000005      .WORD .MD.
2811      ;/RETURN HERE AFTER COMMAND .
2812
2813 010014 052737 000200 001124 BIS #BIT7, $GDDAT ;/EXP'D SR BIT 7 SET.
2814      ;/-MCMD-
2815 010022 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
2816 010026 001424      .WORD KMCSR
2817      .MD.=1
2818      ;/COMMAND=NOP.
2819 010030 000001      .WORD .MD.

```

2820  
2821 010032 117737 171406 001126  
2822 010040 132777 000200 171376  
2823 010046 001004

;/RETURN HERE AFTER COMMAND  
MOVB @BSELS,\$BDDAT ;/READ SR.  
BITB #BIT7,@BSELS ;/DID 7/8 FLAG SET NOW?  
BNE 3\$ ;/IF YES - GOOD.

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2827 010050 104001  
2828

ERROR 1 ;/7/8 FLAG FAILED TO SET IN SIDE 1  
;/AFTER 56. WORDS WRITTEN

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2832 010052 004737 012524  
2833 010056 000424

JSR PC,BINT  
BR TST56 ;;

2834  
2835 010060  
2836 010060 004537 012100  
2837 000004

3\$:

JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
.MD.=4 ;/READ SILO.

2838  
2839 010064 000004

.WORD .MD.  
;/RETURN HERE AFTER COMMAND

2840  
2841  
2842 010066 004537 011656  
2843 010072 001424  
2844 000001

JSR R5,ISSUEC ;/ISSUE COMMAND TO KMC #1.  
.WORD KMCSR  
.MD.=1

2845  
2846 010074 000001

.WORD .MD.  
;/COMMAND=NOP.

2847  
2848 010076 117737 171342 001126  
2849 010104 042737 000200 001124  
2850 010112 132777 000200 171324  
2851 010120 001401

;/RETURN HERE AFTER COMMAND  
MOVB @BSELS,\$BDDAT ;/READ CSR.  
BIC #BIT7,\$GDDAT ;/EXPECT BIT 7 CLEAR.  
BITB #BIT7,@BSELS ;/7/8 FULL FLAG SHOULD BE CLEAR.  
BEQ 4\$ ;/IF CLEAR, GOOD.

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2855 010122 104001  
2856  
2857

ERROR 1 ;/7/8 FLAG ON SIDE 1  
;/FAILED TO CLEAR 56 WORDS WRITTEN  
;/ONE WORD READ; COUNT = 55.

\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2861 010124 004737 012524  
2862  
2863

4\$:

JSR PC,BINT

2864  
2865  
2866 010130 000004  
2867 010132 012737 000010 001160  
2868  
2869

\*\*\*\*\*  
\*TEST 56 \*TEST THAT 7/8 FLAG SETS IN SIDE 2 WHEN 56 WORDS WRITTEN IN SILO  
\*\*\*\*\*

TST56: SCOPE  
MOV #10,\$TIMES ;/DO 10 ITERATIONS  
;/-S78F-

2870 010140 012700 000067  
2871

MOV #55.,R0 ;/SET TO DO 56 TIMES

2872 010144  
2873 010144 004537 012100

1\$:

JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.

```

2874      000001      .MD.=1
2875      ;/COMMAND=NOP.
2876 010150 000001      .WORD .MD.
2877      ;/RETURN HERE AFTER COMMAND
2878 010152 132777 000001 171234 BITB #BIT0,0MBSELS ;/FIFO OUTPUT READY?
2879 010160 001371      BNE 1$
2880 010162 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2881      000005      .MD.=5
2882      ;/WRITE TO $ILO.
2883 010166 000005      .WORD .MD.
2884      ;/RETURN HERE AFTER COMMAND .
2885 010170 005300      DEC R0
2886 010172 001364      BNE 1$ ;/DON@ 55 TIMES?
2887
2888 010174 012737 000054 001124 MOV #54,$GDDAT ;/EXP'D UBSR
2889 010202 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2890      000001      .MD.=1
2891      ;/COMMAND=NOP.
2892 010206 000001      .WORD .MD.
2893      ;/RETURN HERE AFTER COMMAND .
2894 010210 117737 171200 001126 MOVB 0MBSELS,$BDDAT ;/READ SR.
2895 010216 132777 000200 171170 BITB #BIT7,0MBSELS ;/DID 7/8 FLAG SET?
2896 010224 001404      BEQ 2$ ;/NO - GOOD NEED 1 MORE CHARACTER

;*****>> ERROR <<*****
2900 010226 104002      ERROR 2 ;/ERROR 7/8 FLAG SET IN SIDE 2
2901      ;/SET WHEN ONLY 55 WORDS WRITTEN

;*****>> ERROR <<*****
2905 010230 004737 012524 JSR PC,BINT
2906 010234 000450      BR TST57 ;
2907
2908 010236 004537 012100 2$: JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2909 010236 000005      .MD.=5
2910      ;/WRITE TO SILO.
2911
2912 010242 000005      .WORD .MD.
2913      ;/RETURN HERE AFTER COMMAND .
2914
2915 010244 052737 000200 001124 BIS #BIT7,$GDDAT ;/EXP'D SR BIT 7 SET.
2916 010252 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
2917      000001      .MD.=1
2918      ;/COMMAND=NOP.
2919
2920 010256 000001      .WORD .MD.
2921      ;/RETURN HERE AFTER COMMAND .
2922 010260 117737 171130 001126 MOVB 0MBSELS,$BDDAT ;/READ SR.
2923 010266 132777 000200 171120 BITB #BIT7,0MBSELS ;/DID 7/8 FLAG SET NOW?
2924 010274 001004      BNE 3$ ;/IF YES - GOOD.

;*****>> ERROR <<*****
2927 010276 104002      ERROR 2 ;/7/8 FLAG FAILED TO $ET IN SIDE 2

```

2928

;/AFTER 56. WORDS WRITTEN

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2932 010300 004737 012524  
 2933 010304 000424  
 2934  
 2935 010306  
 2936  
 2937 010306 004537 011656  
 2938 010312 001424  
 2939 000004  
 2940  
 2941 010314 000004  
 2942  
 2943 010316 004537 012100  
 2944 000001  
 2945  
 2946 010322 000001  
 2947  
 2948 010324 117737 171064 001126  
 2949 010332 042737 000200 001124  
 2950 010340 132777 000200 171046  
 2951 010346 001401

JSR PC,BINT  
 BR TST57 ;;  
 3\$: ;/-MCMD-  
 ;/ISSUE COMMAND TO KMC #1.  
 JSR R5,ISSUEC  
 .WORD KMCSR  
 .MD.=4 ;/READ SILO.  
 ;/RETURN HERE AFTER COMMAND  
 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.  
 .MD.=1 ;/COMMAND=NOP.  
 .WORD .MD.  
 ;/RETURN HERE AFTER COMMAND  
 MOVB @MBSLS,\$BDDAT ;/READ CSR.  
 BIC #BIT7,\$GDDAT ;/EXPECT BIT 7 CLEAR.  
 BITB #BIT7,@MBSLS ;/7/8 FULL FLAG SHOULD BE CLEAR.  
 BEQ 4\$ ;/IF CLEAR, GOOD.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2955 010350 104002  
2956  
2957

ERROR 2 ;/7/8 FLAG ON SIDE 2  
;/FAILED TO CLEAR 56 WORDS WRITTEN  
;/ONE WORD READ; COUNT = 55.

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

2961 010352 004737 012524  
 2962  
 2963  
 2964  
 2965  
 2966  
 2967 010356 000004  
 2968 010360 012737 000010 001160  
 2969  
 2970  
 2971  
 2972 010366 012700 000100  
 2973  
 2974 010372  
 2975  
 2976 010372 004537 011656  
 2977 010376 001424  
 2978 000001  
 2979  
 2980 010400 000001  
 2981

4\$: JSR PC,BINT  
 ;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
 ;/TEST 57 \*TEST THAT SIDE 1 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRI  
 ;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
 TST57: SCOPE  
 MOV #10,\$TIMES ;;DO 10 ITERATIONS  
 ;/-SOVFL-  
 MOV #64.,R0 ;/SET TO DO 64 XFERRS.  
 1\$: ;/-MCMD-  
 ;/ISSUE COMMAND TO KMC #1.  
 JSR R5,ISSUEC  
 .WORD KMCSR  
 .MD.=1 ;/COMMAND=NOP.  
 .WORD .MD.  
 ;/RETURN HERE AFTER COMMAND .

## E06

MD-11-DRLPN-A  
DRLPN.P11MACY11  
TS7

27(654) 15-DEC-77 08:43 PAGE 57

\*TEST THAT SIDE 1 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRITTEN

SEQ 0069

```

2982 010402 132777 000001 171034 BITB #BIT0, @BSELS ;/FIFO READY?
2983 010410 001370 BNE 1$
2984 ;/-MCMD-
2985 010412 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
2986 010416 001424 .WORD KMCSR
2987 000005 .MD.=5
2988 ;/WRITE TO SILO.
2989 010420 000005 .WORD .MD.
2990 ;/RETURN HERE AFTER COMMAND
2991 010422 005300 DEC R0 ;/DONE 64 TIMES?
2992 010424 001362 BNE 1$ ;/NO - REPEAT
2993 010426 012737 000252 001124 MOV #252, $GDDAT ;/EXP'D IOSR.
2994
2995 ;/-MCMD-
2996 010434 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO kmc #1.
2997 010440 001424 .WORD KMCSR
2998 000001 .MD.=1
2999 ;/COMMAND=NOP.
3000 010442 000001 .WORD .MD.
3001 ;/RETURN HERE AFTER COMMAND
3002 010444 132777 000100 170772 BITB #BIT6, @BSELS ;/FIFO R/W ERROR SET?
3003 010452 001407 BEQ 2$ ;/NO - GOOD
3004 010454 117737 170764 001126 MOV @BSELS, $BDDAT ;/RECORD SR.

;*****>> ERROR <<*****

3008 010462 104001 ERROR 1 ;/FIFO R/W ERROR FLAG SET
3009 ;/IN SIDE 1 WHEN ONLY
3010 ;/64. WORDS WRITTEN

;*****>> ERROR <<*****

3014 010464 004737 012524 JSR PC, BINT
3015 010470 000457 BR TST60 ;;
3016
3017 010472 2$:
3018
3019 010472 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
3020 010476 001424 .WORD KMCSR
3021 000005 .MD.=5
3022 ;/WRITE TO SILO.
3023 010500 000005 .WORD .MD.
3024 ;/RETURN HERE AFTER COMMAND
3025 ;/-MCMD-
3026 010502 004537 011656 JSR R5, ISSUEC ;/ISSUE COMMAND TO KMC #1.
3027 010506 001424 .WORD KMCSR
3028 000001 .MD.=1
3029 ;/COMMAND=NOP.
3030 010510 000001 .WORD .MD.
3031 ;/RETURN HERE AFTER COMMAND
3032 010512 052737 000100 001124 BIS #BIT6, $GDDAT ;/EXPECT BIT 6 TO SET.
3033 010520 117737 170720 001126 MOV @BSELS, $BDDAT ;/READ SR.
3034 010526 132777 000100 170710 BITB #BIT6, @BSELS ;/FIFO OVFL SET?
3035 010534 001007 BNE 3$ ;/YES - GOOD.

```



MD-11-DRLPN-A  
DRLPN.P11MACY11  
T60

27(654)

15-DEC-77

08:43

PAGE 59

\*TEST THAT SIDE 2 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRITTEN

SEQ 0071

```

3090 010652 132777 000001 170534 BITB #BIT0,@MBSELS ;/FIFO READY?
3091 010660 001371 BNE 1$ ;
3092 010662 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3093 000005 .MD.=5 ;/WRITE TO SILO.
3094 .WORD .MD.
3095 010666 000005 ;/RETURN HERE AFTER COMMAND
3096 DEC RD ;/DONE 64 TIMES?
3097 010670 005300 BNE 1$ ;/NO - REPEAT
3098 010672 001364 MOV #256,$GDDAT ;/EXP'D UBSR.
3099 010674 012737 000256 001124 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3100 000001 .MD.=1 ;/COMMAND=NOP.
3101 010702 004537 012100 .WORD .MD.
3102 000001 ;/RETURN HERE AFTER COMMAND
3103 010706 000001 BITB #BIT6,@MBSELS ;/FIFO R/W ERROR SET?
3104 BEQ 2$ ;/NO - GOOD
3105 010710 132777 000100 170476 MOVB @MBSELS,$BDDAT ;/RECORD SR.
3106 010716 001407 BEQ 2$ ;/NO - GOOD
3107 010716 001407 BEQ 2$ ;/NO - GOOD
3108 010720 117737 170470 001126 MOVB @MBSELS,$BDDAT ;/RECORD SR.

;*****>> ERROR <<*****

3112 010726 104002 ERROR 2 ;/FIFO R/W ERROR FLAG SET
3113 ;/IN SIDE 2 WHEN ONLY
3114 ;/64. WORDS WRITTEN

;*****>> ERROR <<*****

3118 010730 004737 012524 JSR PC,BINT
3119 010734 000454 BR TST61 ;;
3120
3121 010736 2$:
3122 010736 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3123 000005 .MD.=5 ;/WRITE TO SILO.
3124 .WORD .MD.
3125 010742 000005 ;/RETURN HERE AFTER COMMAND
3126 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3127 010744 004537 012100 .MD.=1 ;/COMMAND=NOP.
3128 000001
3129 .WORD .MD.
3130 010750 000001 ;/RETURN HERE AFTER COMMAND
3131 BIS #BIT6,$GDDAT ;/EXPECT BIT 6 TO SET.
3132 010752 052737 000100 001124 MOVB @MBSELS,$BDDAT ;/READ SR.
3133 010760 117737 170430 001126 BITB #BIT6,@MBSELS ;/FIFO OVFL SET?
3134 010766 132777 000100 170420 BNE 3$ ;/YES - GOOD.
3135 010774 001007 MOVB @MBSELS,$BDDAT ;/RECORD SR
3136 010776 117737 170412 001126

;*****>> ERROR <<*****

3140 011004 104002 ERROR 2 ;/FIFO R/W ERROR FLAG FAILED TO SET
3141 ;/IN SIDE 2 WHEN 65.
3142 ;/WORDS WRITTEN.

```



H06

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 60  
T60

\*TEST THAT SIDE 2 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRITTEN

SEQ 0072

```

;*****>> ERROR <<*****
3146 011006 004737 012524 JSR PC,BINT
3147 011012 000425 BR TST61 ;;
3148
3149 011014 004737 012524 3$: JSR PC,BINT ;/NOW INITIALIZE.
3150 011020 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3151 000001 .MD.=1
3152 ;/COMMAND=NOP.
3153 011024 000001 .WORD .MD.
3154 ;/RETURN HERE AFTER COMMAND .
3155 011026 042737 000300 001124 BIC #BIT6:BIT7,$GDDAT ;/OVLf AND 7/8 SHOULD CLEAR.
3156 011034 052737 000002 001124 BIS #BIT1,$GDDAT ;/BIT 1 SHOULD SET.
3157 011042 117737 170346 001126 MOVB #MBSEL5,$BDDAT ;/READ SR.
3158 011050 123737 001124 001126 CMPB $GDDAT,$BDDAT ;/EVERYTHING OK?
3159 011056 001401 BEQ 4$ ;YES GET OUT.

;*****>> ERROR <<*****
3163 011060 104002 ERROR 2 ;/FIFO R/W ERROR FLAG FAILED TO
3164 ;/CLEAR SIDE 2 WHEN
3165 ;/INITIALIZED.

;*****>> ERROR <<*****
3169
3170 011062 004737 012524 4$: JSR PC,BINT
3171
3172 ;*****
3173 ;*TEST 61 *TEST BURST MODE XFERR MASTER TO SLAVE
3174 ;*****
3175 011066 000004 TST61: SCOPE
3176 011070 004737 012542 JSR PC,FILBUF ;FILL SEND BUFF WITH RANDOM NUMBERS
3177 011074 10$:
3178 ;/-MCMd-
3179 011074 004537 011656 JSR R5,ISSUEC ;/ISSUE COMMAND TO kMC #1.
3180 011100 001424 .WORD KMCSR
3181 000002 .MD.=2
3182 000202 .MD.=.MD.!200
3183 ;/NPR READ FROM MEM WRITE SILO.
3184 011102 000202 .WORD .MD.
3185 011104 017322 .WORD SENBUF ;/ADDRESS OF MEMORY TO/RO DATA.
3186 011106 000036 .WORD 30. ;/BYTE COUNT FOR XFERR.
3187 ;/RETURN HERE AFTER COMMAND .
3188 011110 012700 017722 MOV #RECBUF,RO
3189 011114 012701 000037 MOV #31.,R1
3190 011120 20$:
3191 011120 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.
3192 000004 .MD.=4
3193 ;/READ SILO.
3194 011124 000004 .WORD .MD.
3195 ;/RETURN HERE AFTER COMMAND .
3196 011126 117720 170256 MOVB #MBSEL3,(0)+
3197 011132 004537 012100 JSR R5,KMSIM ;/DMC SIMULATE KMC INSTR. LIST FOR.

```

MD-11-DRLPN-P  
DRLPN.P11

MACY11  
T61

27(654) 15-DEC-77 08:43 PAGE 61  
\*TEST BURST MODE XFERR MASTER TO SLAVE

SEQ 0073

```

3198      000004      .MD.=4
3199
3200 011136 000004      .WORD .MD.
3201      .:/RETURN HERE AFTER COMMAND .
3202 011140 117720 170244  MOVB  @MBSEL3,(0)+
3203 011144 005301      DEC  R1
3204 011146 001364      BNE  20$
3205 011150 004537 012100  JSR  R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
3206      000001      .MD.=1
3207      .:/COMMAND=NOP.
3208 011154 000001      .WORD .MD.
3209      .:/RETURN HERE AFTER COMMAND .
3210 011156 005037 001126  CLR  $BDDAT
3211 011162 117737 170226 001126  MOVB  @MBSEL5,$BDDAT ;GET CSR DATA
3212 011170 012737 000056 001124  MOV  #56,$GDDAT ;EXPECT 56
3213 011176 123737 001124 001126  CMPB $GDDAT,$BDDAT ;SR OK?
3214 011204 001404      BEQ  1$ ;YES CONTINUE

;*****>> ERROR <<*****
3218 011206 104002      ERROR 2 ;HIGH SPEED OPERATION OF THE
3219 ;SILO'S UP-DOWN COUNTER DIDN'T
3220 ;PRODUCE A NET ZERO RESULT
3221 ;THUS SETTING IOSR BIT 1
3222

;*****>> ERROR <<*****
3226 011210 004737 012524  JSR  PC,BINT
3227 011214 000420      BR   TST62
3228 011216 012700 017722 1$:  MOV  #RECBUF,R0 ;POINT TO RECEIVED DATA
3229 011222 012701 017322  MOV  #SENBUFF,R1 ;POINT TO GOOD DATA
3230 011226 012702 000076  MOV  #62.,R2 ;# OF DATA TO CHECK BYTES
3231 011232 122021 2$:  CMPB (0)+,(1)+ ;DATA GOOD?
3232 011234 001406      BEQ  3$
3233 011236 114037 001126  MOVB -(0),$BDDAT ;NO - PICK UP BAD DATA
3234 011242 114137 001124  MOVB -(1),$GDDAT ;NOW GOOD DATA

;*****>> ERROR <<*****
3238 011246 104006      ERROR 6 ;BUST XFERR THROUGH SILO
3239 ;SIDE 1 TO SIDE 2 PRODUCED
3240 ;A DATA ERROR.

;*****>> ERROR <<*****
3244 011250 000402      BR   TST62
3245 011252 005302      DEC  R2 ;CHECK ALL DATA?
3246 011254 001366      BNE  2$ ;NO - DO NEXT ONE.
3247
3248 ;*****
3249 ;*TEST 62 *TEST BURST MODE XFERR SIDE 2 TO SIDE 1
3250 ;*****
3251 011256 000004      †T62: SCOPE

```

```

3252 011260 004737 012542      JSR      PC,FILBUF      ;FILL SEND BUFF WITH RANDOM NUMBERS
3253                               ;/-MCMD-
3254 011264 004537 011656      JSR      R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
3255 011270 001424              .WORD    KMCSR
3256                               .MD.=3
3257                               ;/READ SILO, NPR WRITE TO MEM.
3258                               .MD.=.MD.!600
3259 011272 000603              .WORD    .MD.
3260 011274 017722              .WORD    RECBUF        ;/ADDRESS OF MEMORY TO/FRO DATA.
3261 011276 000036              .WORD    30.           ;/BYTE COUNT FOR XFERR.
3262                               ;/RETURN HERE AFTER COMMAND
3263 011300 012700 017322      10$:    MOV      #SENBUF,R0  ;ADDR. OF DATA
3264 011304 012701 000076      MOV      #62.,R1        ;#OF WORDS TO z FERR
3265 011310 112077 170074      20$:    MOVB     (0)+,2MBSEL3 ;XFERR BYTE
3266 011314 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
3267                               .MD.=5
3268                               ;/WRITE TO SILO.
3269 011320 000005              .WORD    .MD.
3270                               ;/RETURN HERE AFTER COMMAND
3271 011322 005301              DEC      R1             ;ALL DON#?
3272 011324 001371              BNE     20$            ;NO-CONTINUE
3273 011326 004537 012100      JSR      R5,KMSIM      ;/DMC SIMULATE KMC INSTR. LIST FOR.
3274                               .MD.=1
3275                               ;/COMMAND=NOP.
3276 011332 000001              .WORD    .MD.
3277                               ;/RETURN HERE AFTER COMMAND
3278 011334 005037 001126      CLR     $BDDAT
3279 011340 117737 170100 001126  MOVB     2MBSEL5,$BDDAT ;GET UBSR DATA
3280 011346 012737 000052 001124  MOV      #52,$GDDAT    ;EXPECT 52
3281 011354 123737 001124 001126  CMPB     $GDDAT,$BDDAT ;SR OK?
3282 011362 001404              BEQ     1$             ;YES CONTINUE

;*****>> ERROR <<*****

3286 011364 104001      ERROR 1                ;HIGH SPEED OPERATION OF THE
3287                               ;SILO'S UP-DOWN COUNTER DIDN'T
3288                               ;PRODUCE A NET ZERO RESULT
3289                               ;THUS SETTING UBSR BIT 1
3290                               ;

;*****>> ERROR <<*****

3294 011366 004737 012524      JSR      PC,BINT
3295 011372 000420              BR       TST63
3296 011374 012700 017722      1$:    MOV      #RECBUF,R0  ;POINT TO RECEIVED DATA
3297 011400 012701 017322      MOV      #SENBUF,R1    ;POINT TO GOOD DATA
3298 011404 012702 000076      MOV      #62.,R2       ;# OF DATA TO CHECK BYTES
3299 011410 122021              CMPB     (0)+,(1)+     ;DATA GOOD?
3300 011412 001406              BEQ     3$
3301 011414 114037 001126      MOVB     -(0),$BDDAT   ;NO - PICK UP BAD DATA
3302 011420 114137 001124      MOVB     -(1),$GDDAT   ;NOW GOOD DATA

;*****>> ERROR <<*****

```



```

3360 011606          66$:          MOV      ERCNT,-(SP)      ;;SAVE ERCNT FOR TYPEOUT
3361 011606 013746 001456          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
3362 011612 104405
3363
3364 011614 104401 011622          TYPE      69$          ;;TYPE ASCIZ STRING
3365 011620 000402          BR        68$          ;;GET OVER THE ASCIZ
3366          ;;69$: .ASCIZ /
3367 011626          68$:
3368
3369 011626 013700 000042 $GET42: MOV      @#42,R0      ;;GET MONITOR ADDRESS
3370 011632 001405          BEQ      $DOAGN          ;;BRANCH IF NO MONITOR
3371 011634 000005          RESET          ;;CLEAR THE WORLD
3372 011636 004710 $ENDAD: JSR      PC,(R0)    ;;GO TO MONITOR
3373 011640 000240          NOP            ;;SAVE ROOM
3374 011642 000240          NOP            ;;FOR
3375 011644 000240          NOP            ;;ACT11
3376 011646          $DOAGN:
3377 011646 000137          JMP      @PC+          ;;RETURN
3378 011650 001712 $RTNAD: .WORD   RTNAD
3379 011652          377 000 $ENULL: .BYTE   -1,-1,0      ;;NULL CHARACTER STRING
3380          011656
3381
3382          ;;
3383          ;;*COMMAND ROUTINE
3384          ;;THIS ROUTINE WILL ISSUE COMMANDS TO THE
3385          ;;SPECIFIED KMC-11 AND WAITE FOR COMMAND COMPLETION
3386          ;;UNLESS INHIBITED BY COMMAND CALL.
3387          ;;
3388          ;;CALL= JSR      R5,ISSUEC
3389          ;;
3390          ;;      ARG1
3391          ;;      ARG2
3392          ;;      ARG3      (IF NEEDED)
3393          ;;      ARG4      (IF NEEDED)
3394          ;;      -> RETURNS HERE <-
3395          ;;ARG1 SPECIFIES THE ADDRESS OF THE ADDRESS OF THE KMC-11
3396          ;;ARG2 SPECIFIES THE COMMAND FOR THE KMC-11 FIREMWARE
3397          ;;      IN BITS 0-6
3398          ;;      IN BIT7 WE SPECIFY IF ARG3 AND 4 ARE USED
3399          ;;      IN BIT8 WE SPECIFY IF WE SHOULD WAIT FOR
3400          ;;      COMMAND COMPLETION BEFORE RETURNING
3401          ;;ARG3 SPECIFIES BSEL2+BSEL3 INFO (ADDRESS).
3402          ;;ARG4 SPECIFIES BSEL4 INFO. (BYTE COUNT).
3403          ;;
3404          ;;
3405          ;;
3406          ;;
3407          ;;
3408          ;;
3409          ;;
3410          ;;
3411          ;;
3412          ;;
3413          ;;
          ISSUEC: MOV      R0,-(SP)      ;;SAVE R0
          MOV      @5+,R0      ;;PICK UP KMC ADDR.
          TSTB     (5)          ;;EXTENDED COMMAND?
          BPL      1$          ;;NO-AHEAD.
          MOVB    2(5),2(0)    ;;SET BSEL2
          MOVB    3(5),3(0)    ;;SET BSEL3
          MOVB    4(5),4(0)    ;;SET BSEL4
          BICB    #200,(5)
          MOVB    (5),0(0)    ;;SET COMMAND
          BISB    #200,(5)
          ADD     #6,R5        ;;POINT PAST CALL AND ARGS.
  
```

```

3414 011730 012737 000000 012244      MOV      #0, TIME      ;SET FOR INSTRUCTION TIME OUT.
3415 011736 105765 177773              TSTB     -5(5)         ;IGNOR WAITING FOR DONE?
3416 011742 001040              BNE      ISSEX        ;YES-EXIT.
3417 011744 000404              BR       2$           ;NO-WAIT.
3418
3419 011746 112560 000000      1$:      MOVB     (5)+,0(0) ;ISSUE COMMAND.
3420 011752 105725              TSTB     (5)+         ;IGNOR WAITING FOR DONE?
3421 011754 001033              BNE      ISSEX        ;YES-EXIT.
3422
3423 011756 105710      2$:      TSTB     (0)         ;KMC FINISHED COMMAND?
3424 011760 100433              BMI      ERRIS        ;BIT7 SET IF ERROR.
3425 011762 005237 012244      INC      TIME         ;HAS MICRO-PROC TAKEN TO MUCH TIME?
3426 011766 001373              BNE      2$           ;NO-LOOP.
3427
3428
3429 011770 104401 011776      TYPE     65$         ;:TYPE ASCIZ STRING
3430 011774 000415              BR       64$         ;:GET OVER THE ASCIZ
3431
3432
3433
3434
3435
3436
3437 012044 012600      ;:65$: .ASCIZ <200>#MICRO-HUNG--RESTARTING.#
3438 012046 000205      64$:
3439
3440 012050 122710 000377      MOV      #STACK,SP   ;RESET STACK.
3441 012054 001773              JSR      PC,BINT      ;INITIALIZE MICRO PROCESSORS.
3442 012056 010037 001462      JMP      @SLPADR      ;RESTART SUB TEST.
3443 012062 116037 000005 001126      ISSEX:   MOV      (SP)+,R0 ;RESTORE R0
3444 012070 105037 001127              RTS      R5           ;EXIT
3445
3446
3447
3448 012074 104011      ERRIS:   CMPB     #377,(0)   ;IF CMD=377 THE KMC HAS FINISHED
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467

```

;\*\*\*\*\*>> ERROR <<\*\*\*\*\*  
ERROR 11 ;MICRO PROCESSOR PASSED  
;BACK A DETECTED ERROR  
;SEE TEST WE WERE  
;DOING AT TIME.  
;\*\*\*\*\*>> ERROR <<\*\*\*\*\*

```

BR       ISSEX

;KMSIM +THIS ROUTINE IS USED TO MAKE THE M8200-YC ACT LIKE A
;KMC-11 THE M8200-YC, EVEN THOUGH (P)ROM PROGRAM BLASTED,
;MAY EXECUTE SINGLE INSTRUCTIONS, ONE AT A TIME. WHEN WE
;WILL PASS A NUMBER THAT REPRESENTS THE SEQUENCE OF
;INTRUCTIONS WE WISH IT TO DO. WE THEN TAKE THIS NUMBER
;AND USE IT AS A GUIDE TO START EXECUTING THE CODE
;LOADING INTO THE KMC-11.

```

```

3468                                     ; CALL= JSR R5,KMSIM
3469                                     ; WORD X (COMMAND TO PERFORM)
3470                                     ; RETURNS HERE WHEN DONE
3471
3472 012100 010046 KMSIM: MOV RO,-(SP) ;SAVE RO
3473 012102 010146 MOV RI,-(SP) ;SAVE RI
3474 012104 012500 MOV (5)+,RO ;PICK-UP COMMAND #.
3475 012106 042700 177700 BIC #177700,RO ;STRIP ANY JUNK.
3476 012112 006300 ASL RO ;ADJUST POINTER.
3477 012114 062700 000000G ADD #MRCODE,RO ;ADDR. OF COMMAND LIST
3478 012120 005001 CLR R1
3479 012122 111001 MOVB (0),R1 ;POINTER TO INSTRUCTION LIST.
3480 012124 006301 ASL R1 ;POINTER=POINTER TIMES 2
3481 012126 042701 177000 BIC #177000,R1
3482 012132 062701 000000G ADD #MRCODE,R1 ;PLUS LIST
3483                                     ;POINTER TO LIST OF INSTRUCTIONS
3484                                     ;IS R1++
3485 012136 012100 1$: MOV (1)+,RO ;GET INSTUCTION.
3486 012140 100403 BMI 2$ ;IF A BRANCH INSTRUCTION,EXIT
3487 012142 004737 012166 3$: JSR PC,DOIT ;IF NOT-MAKE DMC-EXECUTE IT.
3488
3489 012146 000773 BR 1$
3490 012150 032761 070000 177776 2$: BIT #070000,-2(1) ;MAKE SURE IT WAS A BR INSTR.
3491 012156 001371 BNE 3$ ;IF A MOVE,GO DO IT.
3492
3493 012160 012601 MOV (SP)+,R1 ;IT WAS A BR RESTORE R1
3494 012162 012600 MOV (SP)+,RO ;AND RO.
3495 012164 000205 RTS ;THEN EXIT.
3496
3497 012166 112777 000002 167210 DOIT: MOVB #BIT01,AMBSEL1 ;SET ROMI
3498 012174 000240 NOP
3499 012176 010077 167214 MOV RO,AMBSEL6 ;LOAD INSTR.
3500 012202 000240 NOP
3501 012204 152777 000003 167172 BISB #BIT1!BIT0,AMBSEL1 ;CLOCK INSTRUCTION
3502 012212 000240 NOP
3503 012214 000240 NOP
3504 012216 000240 NOP
3505 012220 000240 NOP
3506 012222 000240 NOP
3507 012224 000240 NOP
3508 012226 142777 000007 167150 BICB #BIT2!BIT1!BIT0,AMBSEL1 ;CLEAR ROM0,ROMI,SET
3509 012234 000207 RTS PC
3510
3511                                     ;TAKES CARE OF CLOCK INTERRUPTS.
3512 012236 005237 012244 KWINT: INC TIME ;UPDATE TIME.
3513 012242 000002 RTI ;EXIT
3514 012244 000000 TIME: .WORD 0 ;SET TO MINUS NUMBER BY USER.
3515
3516                                     ;*
3517                                     ;*LOAD THIS ROUTINE LOADS THE UCODE
3518                                     ;* INTO THE KMC-11 UPROCESSOR.
3519                                     ;* CALL= JSR R5,LOAD
3520                                     ;*
3521                                     ;*MAKE SURE JUMPER W1 IS REMOVED ON 8254 MODULE!!!
                                     ;*

```

3522	012246	005004	LOAD:	CLR	R4	;CLEAR ERROR TIMER
3523	012250	012700	3\$:	MOV	#MRCODE,RO	;GET ADDRESS OF UCODE
3524						;CLEAR CSRS.



```

3525 012254 005077 167144 CLR @KMCSR ;SET ADDR 0
3526 ; ;IN BOTH KMC-11S
3527 012260 005077 167142 CLR @KMADR ;SELECT CRAM.
3528 ; ;WRITE DATA
3529 012264 052777 002000 167132 1$: BIS #2000,@KMCSR ;SET CRAM WRITE
3530 ; ;NOTE: IF PROGRAM SEEMS TO STOP IN THIS AREA,
3531 012272 012077 167132 MOV (0)+,@KMDBR ;MAKE SURE JUMPER W1 ON 8254
3532 ; ;MODULE IS REMOVED!!!
3533 012276 052777 020000 167120 BIS #20000,@KMCSR ;DISABLE CRAM
3534 ; ;IN UNIT
3535 ; ;DONE ALL UCODE?
3536 ; ;NO - DO NEXT INSTR.
3537 ; ;NOW LETS CHECK TO MAKE SURE
3538 012304 005077 167114 CLR @KMCSR ;UCODE GOT LOADED CORRECTLY.
3539 012310 005277 167112 INC @KMADR
3540 012314 020027 000000G CMP RO,#UCODEE
3541 012320 001361 BNE 1$
3542 ; ;
3543 ; ;
3544 012322 005077 167100 CLR @KMADR ;NOW LETS CHECK TO MAKE SURE
3545 012326 012700 000000G MOV #MRCODE,RO ;UCODE GOT LOADED CORRECTLY.
3546 ; ;
3547 012332 052777 002000 167064 2$: BIS #2000,@KMCSR ;SELECT CRAM.
3548 012340 013737 001426 001462 MOV KMADR,KLAD
3549 012346 022077 167056 CMP (0)+,@KMDBR ;DATA OK?
3550 012352 001013 BNE LOADER
3551 012354 005077 167044 CLR @KMCSR ;CLEAR THE CSR
3552 012360 005277 167042 INC @KMADR ;CHANGE UCODE ADDR.
3553 012364 020027 000000G CMP RO,#UCODEE ;AT END?
3554 012370 001360 BNE 2$
3555 012372 012777 100000 167024 MOV #BIT15,@KMCSR ;SET RUN.
3556 012400 000205 RTS R5 ;RETURN.
3557 ; ;
3558 ; ;
3559 ; ;
3560 ; ;
3561 ; ;
3562 ; ;
3563 ; ;
3564 012402 012402 104401 012414 ;COME HERE ON LOAD ERROR
3565 012404 105204 000415 LOADER=. INCB R4 ;MAKE SURE WE DON'T TRY TOO MANY TIMES.
3566 012406 100321 BPL 3$ ;LOOP BACK TRY AGAIN.
3567 012412 000415 TYPE ,65$ ;;TYPE ASCIZ STRING
3568 012446 104401 012454 BR ,64$ ;;GET OVER THE ASCIZ
3569 012452 000404 ;;65$: .ASCIZ <200>!ERROR IN LOADING UCODE !
3570 012464 104401 012454 BR ,66$ ;;TYPE ASCIZ STRING
3571 012466 000404 ;;66$: .ASCIZ <200>!GOOD= !
3572 012466 005740 ;;67$: .ASCIZ <200>!GOOD= !
3573 012466 005740 TST -(0)
3574 012466 011046 10$: MOV (RO),-(SP) ;;SAVE (RO) FOR TYPEOUT
3575 012470 104402 TYPOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3576 012472 104401 012500 TYPE ,69$ ;;TYPE ASCIZ STRING
3577 012476 000404 BR ,68$ ;;GET OVER THE ASCIZ
3578 ;;69$: .ASCIZ ! BAD= !

```

```

3579 012510          68$:      MOV      2KMDBR,-(SP)      ;;SAVE 2KMDBR FOR TYPEOUT
3580 012510 017746 166714      TYPOC      ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3581 012514 104402      HALT      ;;ERROR IN LOADING UCODE,RETRY
3582 012516 000000      ;;HAVE FAILED. RUN KMC11 DIAGNOSTIC.
3583
3584 012520 000137 012250      JMP      3$
3585
3586
3587      ;;
3588      ;;THIS ROUTINE WILL ISSUE INITIALIZE TO
3589      ;;BOTH KMC11S.
3590      ;;CALL= JSR      PC,BINT
3591
3592 012524 000005          BINT:     RESET
3593 012526 012777 100000 166670      MOV      #BIT15,2KMCSR      ;START #1.
3594 012534 005077 166634      CLR      2KMMCSR      ;HALT #2.
3595 012540 000207      RTS      PC
3596
3597      ;;
3598      ;;ROUTINE TO FILL SENBUF WITH RANDOM
3599      ;;DATA
3600
3601 012542 012700 017322          FILBUF:  MOV      #SENBUF,RO      ;GET ADDRESS
3602
3603 012546 004737 012600          1$:      JSR      PC,RAND      ;GET 2 RANDOM NUMBERS
3604 012552 013720 012626      MOV      RANA,(0)+
3605 012556 020027 017722      CMP      RO,#RECBUF      ;BUFFER FULL?
3606 012562 001405      BEQ      FILEX
3607 012564 013720 012630      MOV      RANB,(0)+
3608 012570 020027 017722      CMP      RO,#RECBUF      ;BUFFER FULL?
3609 012574 001364      BNE      1$
3610 012576 000207          FILEX:   RTS      PC
3611
3612 012600 063737 012626 012630          RAND:   ADD      RANA,RANB
3613 012606 005537 012626      ADC      RANA
3614 012612 063737 012630 012626      ADD      RANB,RANA
3615 012620 005537 012630      ADC      RANB
3616 012624 000207      RTS      PC
3617
3618 012626 123456          RANA:   123456
3619 012630 071234          RANB:   071234
3620
3621      ;;ROUTINE TO READ/MODIFY CRAM.
3622      ;;L&S AT 210
3623
3624 012632          MOCRAM:
3625 012632 104401 012640      TYPE      65$      ;;TYPE ASCIZ STRING
3626 012636 000406      BR      64$      ;;GET OVER THE ASCIZ
3627
3628 012654          ;;65$:  .ASCIZ  <200>#KMC ADR? #
3629 012654          64$:
3630 012656 012600      RDOCT
3631 012660 052700 160000      MOV      (SP)+,RO
3632 012664 104401 012672      BIS      #160000,RO
3633
3634          1$:      TYPE      ,67$      ;;TYPE ASCIZ STRING
    
```

3633	012670	000407					
3634				;;67\$:	BR	66\$	::GET OVER THE ASCIZ
3635	012710			66\$:	.ASCIZ	<200>#CRAM ADDR?#	
3636	012710	104412			RDOCT		
3637	012712	012601			MOV	(SP)+,R1	
3638	012714			9\$:			
3639	012714	104401	012722		TYPE	69\$	::TYPE ASCIZ STRING
3640	012720	000401			BR	68\$	::GET OVER THE ASCIZ
3641				;;69\$:	.ASCIZ	<200>##	
3642	012724			68\$:			
3643	012724	010146			MOV	R1,-(SP)	::SAVE R1 FOR TYPEOUT
3644	012726	104402			TYPOC		::GO TYPE--OCTAL ASCII(ALL DIGITS)
3645	012730	104401	012736		TYPE	71\$	::TYPE ASCIZ STRING
3646	012734	000402			BR	70\$	::GET OVER THE ASCIZ
3647				;;71\$:	.ASCIZ	#/#	
3648	012742			70\$:			
3649	012742	005010			CLR	(0)	

3650	012744	010160	000004		MOV	R1,4(0)		::SET ADDR.
3651	012750	052710	002000		BIS	#2000,(0)		::SELECT CRAM.
3652	012754	016002	000006		MOV	6(0),R2		::READ CRAM.
3653	012760	010246			MOV	R2,-(SP)		::SAVE R2 FOR TYPEOUT
3654	012762	104402			TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
3655	012764	104401	012772		TYPE	73\$		::TYPE ASCIZ STRING
3656	012770	000403			BR	72\$		::GET OVER THE ASCIZ
3657				::73\$:	.ASCIZ	# \C\ #		
3658	013000			72\$:				
3659	013000	105777	166140	10\$:	TSTB	2\$TKS		
3660	013004	100375			BPL	10\$		
3661	013006	117703	166134		MOVB	2\$TKB,R3		
3662	013012	042703	000200		BIC	#200,R3		
3663	013016	120327	000012		CMPB	R3,#12		
3664	013022	001002			BNE	11\$		
3665	013024	005201			INC	R1		
3666	013026	000732			BR	9\$		
3667	013030	120327	000015	11\$:	CMPB	R3,#15		
3668	013034	001713			BEQ	1\$		
3669								
3670	013036	104401	013044		TYPE	75\$		::TYPE ASCIZ STRING
3671	013042	000402			BR	74\$		::GET OVER THE ASCIZ
3672				::75\$:	.ASCIZ	# / #		
3673	013050			74\$:				
3674	013050	104412			RDOCT			
3675	013052	012602			MOV	(SP)+,R2		;NEW DATA
3676	013054	001703			BEQ	1\$		;=0,NO NEW DATA
3677	013056	010260	000006		MOV	R2,6(0)		
3678	013062	052710	020000		BIS	#20000,(0)		;CRAM WRITE.
3679	013066	000676			BR	1\$		
3680								
3681								
3682								
3683	013070			ODT:				
3684	013070	104401	013076		TYPE	65\$		::TYPE ASCIZ STRING
3685	013074	000404			BR	64\$		::GET OVER THE ASCIZ
3686				::65\$:	.ASCIZ	<200>#ADDR? #		
3687	013106			64\$:				
3688	013106	104412			RDOCT			
3689								
3690	013110	012600			MOV	(SP)+,R0		
3691	013112			1\$:				
3692	013112	104401	013120		TYPE	67\$		::TYPE ASCIZ STRING
3693	013116	000401			BR	66\$		::GET OVER THE ASCIZ
3694				::67\$:	.ASCIZ	<200>##		
3695	013122			66\$:				
3696	013122	010046			MOV	R0,-(SP)		::SAVE R0 FOR TYPEOUT
3697	013124	104402			TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)
3698	013126	104401	013134		TYPE	69\$		::TYPE ASCIZ STRING
3699	013132	000402			BR	68\$		::GET OVER THE ASCIZ
3700				::69\$:	.ASCIZ	#/ #		
3701	013140			68\$:				
3702	013140	011046			MOV	(0),-(SP)		::SAVE (0) FOR TYPEOUT
3703	013142	104402			TYPOC			::GO TYPE--OCTAL ASCII(ALL DIGITS)

```

3704 013144
3705 013144 104401 013152
3706 013150 000402
3707
3708 013156
3709 013156 105777 165762
3710 013162 100375
3711 013164 117701 165756
3712 013170 042701 000200
3713 013174 122701 000015
3714 013200 001733
3715 013202 122701 000012
3716 013206 001003
3717 013210 062700 000002
3718 013214 000736
3719 013216 122701 000033
3720 013222 001003
3721 013224 162700 000002
3722 013230 000730
3723 013232
3724 013232 104401 013240
3725 013236 000403
3726
3727 013246
3728 013246 104412
3729 013250 012610
3730 013252 000734
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757 013254 017646 000000

```

```

5$:
TYPE 71$
BR 70$
;;TYPE ASCIZ STRING
;;GET OVER THE ASCIZ
71$: .ASCIZ #\?\#
70$:
2$: TSTB 2$TKS
BPL 2$
MOVB 2$TKB,R1
BIC #200,R1
CMPB #15,R1
BEQ 0DT
CMPB #12,R1
BNE 3$
ADD #2,RO
BR 1$
3$: CMPB #33,R1
BNE 4$
SUB #2,RO
BR 1$
4$:
TYPE 73$
BR 72$
;;TYPE ASCIZ STRING
;;GET OVER THE ASCIZ
73$: .ASCIZ #NEW=#
72$:
RDOCT
MOV (SP)+,(O)
BR 5$

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE
*****
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
*OCTAL (ASCII) NUMBER AND TYPE IT.
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
*CALL:
* MOV NUM,-(SP) ;;NUMBER TO BE TYPED
* TYPOS ;;CALL FOR TYPEOUT
* .BYTE N ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
* .BYTE M ;;M=1 OR 0
* ;;1=TYPE LEADING ZEROS
* ;;0=SUPPRESS LEADING ZEROS
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
*$TYPOS OR $TYPOC
*CALL:
* MOV NUM,-(SP) ;;NUMBER TO BE TYPED
* TYPON ;;CALL FOR TYPEOUT
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
*CALL:
* MOV NUM,-(SP) ;;NUMBER TO BE TYPED
* TYPOC ;;CALL FOR TYPEOUT
*$TYPOS: MOV 2(SP),-(SP) ;;PICKUP THE MODE

```

```

3758 013260 116637 000001 013477      MOVB      1(SP), $OFILL      ;; LOAD ZERO FILL SWITCH
3759 013266 112637 013501          MOVB      (SP)+, $OMODE+1  ;; NUMBER OF DIGITS TO TYPE
3760 013272 062716 000002          ADD       #2, (SP)        ;; ADJUST RETURN ADDRESS
3761 013276 000406          BR        $TYPON
3762 013300 112737 000001 013477  $TYPON: MOVB      #1, $OFILL      ;; SET THE ZERO FILL SWITCH
3763 013306 112737 000006 013501          MOVB      #6, $OMODE+1    ;; SET FOR SIX(6) DIGITS
3764 013314 112737 000005 013476  $TYPON: MOVB      #5, $OCNT      ;; SET THE ITERATION COUNT
3765 013322 010346          MOV       R3, -(SP)      ;; SAVE R3
3766 013324 010446          MOV       R4, -(SP)      ;; SAVE R4
3767 013326 010546          MOV       R5, -(SP)      ;; SAVE R5
3768 013330 113704 013501          MOVB      $OMODE+1, R4    ;; GET THE NUMBER OF DIGITS TO TYPE
3769 013334 005404          NEG       R4
3770 013336 062704 000006          ADD       #6, R4          ;; SUBTRACT IT FOR MAX. ALLOWED
3771 013342 110437 013500          MOVB      R4, $OMODE      ;; SAVE IT FOR USE
3772 013346 113704 013477          MOVB      $OFILL, R4      ;; GET THE ZERO FILL SWITCH
3773 013352 016605 000012          MOV       12(SP), R5     ;; PICKUP THE INPUT NUMBER
3774 013356 005003          CLR       R3             ;; CLEAR THE OUTPUT WORD
3775 013360 006105          1$:      ROL       R5      ;; ROTATE MSB INTO "C"
3776 013362 000404          BR        3$
3777 013364 006105          2$:      ROL       R5      ;; GO DO MSB
3778 013366 006105          ROL       R5             ;; FORM THIS DIGIT
3779 013370 006105          ROL       R5
3780 013372 010503          MOV       R5, R3
3781 013374 006103          3$:      ROL       R3      ;; GET LSB OF THIS DIGIT
3782 013376 105337 013500          DECB     $OMODE          ;; TYPE THIS DIGIT?
3783 013402 100016          BPL      7$             ;; BR IF NO
3784 013404 042703 177770          BIC      #177770, R3     ;; GET RID OF JUNK
3785 013410 001002          BNE      4$             ;; TEST FOR 0
3786 013412 005704          TST      R4             ;; SUPPRESS THIS 0?
3787 013414 001403          BEQ      5$             ;; BR IF YES
3788 013416 005204          4$:      INC       R4      ;; DON'T SUPPRESS ANYMORE 0'S
3789 013420 052703 000060          BIS      #'0, R3        ;; MAKE THIS DIGIT ASCII
3790 013424 052703 000040          BIS      #' , R3        ;; MAKE ASCII IF NOT ALREADY
3791 013430 110337 013474          MOVB     R3, #8$        ;; SAVE FOR TYPING
3792 013434 104401 013474          TYPE     #8$           ;; GO TYPE THIS DIGIT
3793 013440 105337 013476          7$:      DECB     $OCNT      ;; COUNT BY 1
3794 013444 003347          BGT      2$             ;; BR IF MORE TO DO
3795 013446 002402          BLT      6$             ;; BR IF DONE
3796 013450 005204          INC      R4             ;; INSURE LAST DIGIT ISN'T A BLANK
3797 013452 000744          BR        2$           ;; GO DO THE LAST DIGIT
3798 013454 012605          6$:      MOV      (SP)+, R5   ;; RESTORE R5
3799 013456 012604          MOV      (SP)+, R4   ;; RESTORE R4
3800 013460 012603          MOV      (SP)+, R3   ;; RESTORE R3
3801 013462 016666 000002 000004          MOV      2(SP), 4(SP)  ;; SET THE STACK FOR RETURNING
3802 013470 012616          MOV      (SP)+, (SP)
3803 013472 000002          RTI
3804 013474 000          8$:      .BYTE    0          ;; RETURN
3805 013475 000          .BYTE    0          ;; STORAGE FOR ASCII DIGIT
3806 013476 000          $OCNT:   .BYTE    0          ;; TERMINATOR FOR TYPE ROUTINE
3807 013477 000          $OFILL:  .BYTE    0          ;; OCTAL DIGIT COUNTER
3808 013500 000000          $OMODE:  .WORD    0          ;; ZERO FILL SWITCH
3809          .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
3810
3811
;:*****

```

```

3812 ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
3813 ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
3814 ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
3815 ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
3816 ;*REPLACED WITH SPACES.
3817 ;*CALL:
3818 ;*      MOV      NUM,-(SP)      ;; PUT THE BINARY NUMBER ON THE STACK
3819 ;*      TYPDS    ;; GO TO THE ROUTINE
3820
3821 $TYPDS:
3822      MOV      R0,-(SP)      ;; PUSH R0 ON STACK
3823      MOV      R1,-(SP)      ;; PUSH R1 ON STACK
3824      MOV      R2,-(SP)      ;; PUSH R2 ON STACK
3825      MOV      R3,-(SP)      ;; PUSH R3 ON STACK
3826      MOV      R5,-(SP)      ;; PUSH R5 ON STACK
3827      MOV      #20200,-(SP)  ;; SET BLANK SWITCH AND SIGN
3828      MOV      20(SP),R5    ;; GET THE INPUT NUMBER
3829      BPL      1$           ;; BR IF INPUT IS POS.
3830      NEG      R5           ;; MAKE THE BINARY NUMBER POS.
3831      MOVVB   #'-,1(SP)    ;; MAKE THE ASCII NUMBER NEG.
3832      CLR      R0           ;; ZERO THE CONSTANTS INDEX
3833      MOV      #DBLK,R3    ;; SETUP THE OUTPUT POINTER
3834      MOVVB   #' ,(R3)+    ;; SET THE FIRST CHARACTER TO A BLANK
3835      CLR      R2           ;; CLEAR THE BCD NUMBER
3836      MOV      $DTB1(R0),R1 ;; GET THE CONSTANT
3837      SUB      R1,R5       ;; FORM THIS BCD DIGIT
3838      BLT     4$           ;; BR IF DONE
3839      INC      R2           ;; INCREASE THE BCD DIGIT BY 1
3840      BR      3$
3841      ADD     R1,R5        ;; ADD BACK THE CONSTANT
3842      TST     R2           ;; CHECK IF BCD DIGIT=0
3843      BNE     5$          ;; FALL THROUGH IF 0
3844      TSTB   (SP)         ;; STILL DOING LEADING 0'S?
3845      BMI     7$          ;; BR IF YES
3846      ASLB   (SP)         ;; MSD?
3847      BCC     6$          ;; BR IF NO
3848      MOVVB   1(SP),-1(R3) ;; YES--SET THE SIGN
3849      BIS     #'0,R2      ;; MAKE THE BCD DIGIT ASCII
3850      BIS     #' ,R2      ;; MAKE IT A SPACE IF NOT ALREADY A DIGIT
3851      MOVVB   R2,(R3)+    ;; PUT THIS CHARACTER IN THE OUTPUT BUFFER
3852      TST     (R0)+       ;; JUST INCREMENTING
3853      CMP     R0,#10      ;; CHECK THE TABLE INDEX
3854      BLT     2$          ;; GO DO THE NEXT DIGIT
3855      BGT     8$          ;; GO TO EXIT
3856      MOV     R5,R2       ;; GET THE LSD
3857      BR      6$          ;; GO CHANGE TO ASCII
3858      TSTB   (SP)+        ;; WAS THE LSD THE FIRST NON-ZERO?
3859      BPL     9$          ;; BR IF NO
3860      MOVVB   -1(SP),-2(R3) ;; YES--SET THE SIGN FOR TYPING
3861      CLRB   (R3)         ;; SET THE TERMINATOR
3862      MOV     (SP)+,R5     ;; POP STACK INTO R5
3863      MOV     (SP)+,R3     ;; POP STACK INTO R3
3864      MOV     (SP)+,R2     ;; POP STACK INTO R2
3865      MOV     (SP)+,R1     ;; POP STACK INTO R1

```





```

3920 014072 000002
3921
3922
3923
3924
3925
3926
3927
3928 014074
3929 014074 104401 001167
3930 014100 010046
3931 014102 005000
3932 014104 153700 001114
3933 014110 001004
3934
3935 014112 013746 001116
3936
3937 014116 104402
3938 014120 000426
3939 014122 005300
3940 014124 006300
3941 014126 006300
3942 014130 006300
3943 014132 062700 001254
3944 014136 012037 014146
3945 014142 001404
3946 014144 104401
3947 014146 000000
3948 014150 104401 001167
3949 014154 012037 014164
3950 014160 001404
3951 014162 104401
3952 014164 000000
3953 014166 104401 001167
3954 014172 011000
3955 014174 001004
3956 014176 012600
3957 014200 104401 001167
3958 014204 000207
3959 014206
3960 014206 013046
3961 014210 104402
3962 014212 005710
3963 014214 001770
3964 014216 104401 014224
3965 014222 000771
3966 014224 020040 000
3967 014230
3968
3969
3970
3971
3972
3973
    
```

```

RTI
.SBTTL ERROR MESSAGE TYPEOUT ROUTINE
;*****
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.
$ERRTYP:
TYPE $SCLF ;;"CARRIAGE RETURN" & "LINE FEED"
MOV RO,-(SP) ;SAVE RO
CLR RO ;PICKUP THE ITEM INDEX
BISB @#$ITEMB,RO
BNE 1$ ;IF ITEM NUMBER IS ZERO, JUST
;TYPE THE PC OF THE ERROR
MOV $ERRPC,-(SP) ;SAVE $ERRPC FOR TYPEOUT
;ERROR ADDRESS
TYPOC ;GO TYPE--OCTAL ASCII(ALL DIGITS)
BR 6$ ;GET OUT
1$: DEC RO ;ADJUST THE INDEX SO THAT IT WILL
ASL RO ;WORK FOR THE ERROR TABLE
ASL RO
ASL RO
ADD #$ERRTB,RO ;FORM TABLE POINTER
MOV (RO)+,2$ ;PICKUP "ERROR MESSAGE" POINTER
BEQ 3$ ;SKIP TYPEOUT IF NO POINTER
TYPE ;TYPE THE "ERROR MESSAGE"
WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
2$: TYPE $SCLF ;;"CARRIAGE RETURN" & "LINE FEED"
MOV (RO)+,4$ ;PICKUP "DATA HEADER" POINTER
BEQ 5$ ;SKIP TYPEOUT IF 0
TYPE ;TYPE THE "DATA HEADER"
WORD 0 ;"DATA HEADER" POINTER GOES HERE
3$: TYPE $SCLF ;;"CARRIAGE RETURN" & "LINE FEED"
MOV (RO),RO ;PICKUP "DATA TABLE" POINTER
BNE 7$ ;GO TYPE THE DATA
MOV (SP)+,RO ;RESTORE RO
TYPE $SCLF ;;"CARRIAGE RETURN" & "LINE FEED"
RTS PC ;RETURN
7$: MOV @ (RO)+,-(SP) ;SAVE @ (RO)+ FOR TYPEOUT
TYPOC ;GO TYPE--OCTAL ASCII(ALL DIGITS)
TST (RO) ;IS THERE ANOTHER NUMBER?
BEQ 6$ ;BR IF NO
TYPE 8$ ;TYPE TWO(2) SPACES
BR 7$ ;LOOP
8$: .ASCIZ / / ;TWO(2) SPACES
.EVEN
.SBTTL SCOPE HANDLER ROUTINE
;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER($STNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
    
```

```

3974 ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3975 ;*SW14=1 LOOP ON TEST
3976 ;*SW11=1 INHIBIT ITERATIONS
3977 ;*SW08=1 LOOP ON TEST IN SWR<7:0>
3978 ;*CALL
3979 ;*
3980 ;* SCOPE ;;SCOPE=IOT
3981 $SCOPE:
3982 014230 104407 CKSWR ;;TEST FOR CHANGE IN SOFT-SWR
3983 014232 104407 CKSWR
3984 014234 032777 040000 164676 1$: BIT #BIT14,@SWR ;;LOOP ON PRESENT TEST?
3985 014242 001070 BNE $OVER ;;YES IF SW14=1
3986 ;*#####START OF CODE FOR THE XOR TESTER#####
3987 014244 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
3988 ;* THIS INSTRUCTION TO A "NOP" (NOP=240)
3989 014246 013746 000004 MOV @ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
3990 014252 012737 014272 000004 MOV #5$,@ERRVEC ;;SET FOR TIMEOUT
3991 014260 005737 177060 TST @177060 ;;TIME OUT ON XOR?
3992 014264 012637 000004 MOV (SP)+,@ERRVEC ;;RESTORE THE ERROR VECTOR
3993 014270 000446 BR $SVLAD ;;GO TO THE NEXT TEST
3994 014272 022626 000004 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
3995 014274 012637 000004 MOV (SP)+,@ERRVEC ;;RESTORE THE ERROR VECTOR
3996 014300 000451 BR $OVER ;;LOOP ON THE PRESENT TEST
3997 014302 6$:;#####END OF CODE FOR THE XOR TESTER#####
3998 014302 032777 000400 164630 BIT #BIT08,@SWR ;;LOOP ON SPEC. TEST?
3999 014310 001404 BEQ 2$ ;;BR IF NO
4000 014312 127737 164622 001102 CMPB @SWR,$STSTNM ;;ON THE RIGHT TEST? SWR<7:0>
4001 014320 001441 BEQ $OVER ;;BR IF YES
4002 014322 105737 001103 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
4003 014326 001404 BEQ 3$ ;;BR IF NO
4004 014330 105037 001103 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
4005 014334 005037 001160 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
4006 014340 032777 004000 164572 3$: BIT #BIT11,@SWR ;;INHIBIT ITERATIONS?
4007 014346 001011 BNE 1$ ;;BR IF YES
4008 014350 005737 001200 TST $PASS ;;IF FIRST PASS OF PROGRAM
4009 014354 001406 BEQ 1$ ;;INHIBIT ITERATIONS
4010 014356 005237 001104 INC $ICNT ;;INCREMENT ITERATION COUNT
4011 014362 023737 001160 001104 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
4012 014370 002015 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
4013 014372 012737 000001 001104 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
4014 014400 013737 014440 001160 MOV $SMXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
4015 014406 105237 001102 $SVLAD: INCB $STSTNM ;;COUNT TEST NUMBERS
4016 014412 113737 001102 001176 MOVB $STSTNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
4017 014420 011637 001106 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
4018 014424 013777 001102 164510 $OVER: MOV $STSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
4019 014432 013716 001106 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
4020 014436 000002 RTI ;;FIXES PS
4021 014440 001000 $SMXCNT: 512. ;;MAX. NUMBER OF ITERATIONS
4022 ;.SBTTL TTY INPUT ROUTINE
4023 ;
4024 ;:*****
4025 ;.ENABL LSB
4026 ;
4027 ;:*****

```

```

4028 ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
4029 ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
4030 ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
4031 ;*WHEN OPERATING IN TTY FLAG MODE.
4032 014442 022737 000176 001140 $CKSWR: CMP      #SWREG, SWR      ;; IS THE SOFT-SWR SELECTED?
4033 014450 001074          BNE      15$          ;; BRANCH IF NO
4034 014452 105777 164466          TSTB   @STKS          ;; CHAR THERE?
4035 014456 100071          BPL     15$          ;; IF NO, DON'T WAIT AROUND
4036 014460 117746 164462          MOVB  @STKB, -(SP)    ;; SAVE THE CHAR
4037 014464 042716 177600          BIC   #1C177, (SP)   ;; STRIP-OFF THE ASCII
4038 014470 022726 000007          CMP   #7 (SP)+       ;; IS IT A CONTROL G?
4039 014474 001062          BNE   15$          ;; NO, RETURN TO USER
4040 014476 123727 001134 000001          CMPB  $AUTOB, #1    ;; ARE WE RUNNING IN AUTO-MODE?
4041 014504 001456          BEQ   15$          ;; BRANCH IF YES
4042
4043 014506 104401 015167          TYPE  ,SCNTLG       ;; ECHO THE CONTROL-G (↑G)
4044 014512 104401 015174          TYPE  ,SMSWR       ;; TYPE CURRENT CONTENTS
4045 014516 013746 000176          MOV   SWREG, -(SP)  ;; SAVE SWREG FOR TYPEOUT
4046 014522 104402          TYPOC          ;; GO TYPE--OCTAL ASCII(ALL DIGITS)
4047 014524 104401 015205          TYPE  ,SMNEW       ;; PROMPT FOR NEW SWR
4048 014530 005046          19$: CLR   -(SP)      ;; CLEAR COUNTER
4049 014532 005046          CLR   -(SP)      ;; THE NEW SWR
4050 014534 105777 164404          7$: TSTB  @STKS      ;; CHAR THERE?
4051 014540 100375          BPL   7$          ;; IF NOT TRY AGAIN
4052
4053 014542 117746 164400          MOVB  @STKB, -(SP)  ;; PICK UP CHAR
4054 014546 042716 177600          BIC   #1C177, (SP) ;; MAKE IT 7-BIT ASCII
4055
4056
4057
4058 014552 021627 000025          9$:  CMP   (SP), #25   ;; IS IT A CONTROL-U?
4059 014556 001005          BNE   10$         ;; BRANCH IF NOT
4060 014560 104401 015162          TYPE  ,SCNTLU      ;; YES, ECHO CONTROL-U (↑U)
4061 014564 062706 000006          20$: ADD   #6, SP     ;; IGNORE PREVIOUS INPUT
4062 014570 000757          BR    19$         ;; LET'S TRY IT AGAIN
4063
4064
4065 014572 021627 000015          10$: CMP   (SP), #15   ;; IS IT A <CR>?
4066 014576 001022          BNE   16$         ;; BRANCH IF NO
4067 014600 005766 000004          TST   4(SP)       ;; YES, IS IT THE FIRST CHAR?
4068 014604 001403          BEQ   11$         ;; BRANCH IF YES
4069 014606 016677 000002 164324          MOV   2(SP), @SWR  ;; SAVE NEW SWR
4070 014614 062706 000006          ADD   #6, SP      ;; CLEAR UP STACK
4071 014620 104401 001167          11$: TYPE  ,SCRLF    ;; ECHO <CR> AND <LF>
4072 014624 123727 001135 000001          14$: CMPB  $INTAG, #1  ;; RE-ENABLE TTY KBD INTERRUPTS?
4073 014632 001003          BNE   15$         ;; BRANCH IF NOT
4074 014634 012777 000100 164302          MOV   #100, @STKS ;; RE-ENABLE TTY KBD INTERRUPTS
4075 014642 000002          RTI                    ;; RETURN
4076 014644 004737 015532          15$: JSR   PC, $TYPEC  ;; ECHO CHAR
4077 014650 021627 000060          16$: CMP   (SP), #60  ;; CHAR < 0?
4078 014654 002420          BLT   18$         ;; BRANCH IF YES
4079 014656 021627 000067          CMP   (SP), #67  ;; CHAR > 7?
4080 014662 003015          BGT   18$         ;; BRANCH IF YES
4081 014664 042726 000060          BIC   #60, (SP)+  ;; STRIP-OFF ASCII

```

```

4082 014670 005766 000002      TST      2(SP)      ;; IS THIS THE FIRST CHAR
4083 014674 001403      BEQ      17$      ;; BRANCH IF YES
4084 014676 006316      ASL      (SP)      ;; NO, SHIFT PRESENT
4085 014700 006316      ASL      (SP)      ;; CHAR OVER TO MAKE
4086 014702 006316      ASL      (SP)      ;; ROOM FOR NEW ONE.
4087 014704 005266 000002      17$: INC      2(SP)      ;; KEEP COUNT OF CHAR
4088 014710 056616 177776      BIS      -2(SP), (SP) ;; SET IN NEW CHAR
4089 014714 000707      BR       7$      ;; GET THE NEXT ONE
4090 014716 104401 001166      18$: TYPE   $QUES    ;; TYPE ?<CR><LF>
4091 014722 000720      BR       20$    ;; SIMULATE CONTROL-U
4092
4093      .DSABL  LSB
4094
4095      ;*****
4096      ;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
4097      ;CALL:
4098      ; RDCHR      ;; INPUT A SINGLE CHARACTER FROM THE TTY
4099      ; RETURN HERE ;; CHARACTER IS ON THE STACK
4100      ;
4101      ;
4102      ;
4103 014724 011646      $RDCHR: MOV      (SP), -(SP) ;; PUSH DOWN THE PC
4104 014726 016666 000004 000002      MOV      4(SP), 2(SP) ;; SAVE THE PS
4105 014734 105777 164204      1$: TSTB   $STKS    ;; WAIT FOR
4106 014740 100375      BPL      1$      ;; A CHARACTER
4107 014742 117766 164200 000004      MOVB   $STKB, 4(SP) ;; READ THE TTY
4108 014750 042766 177600 000004      BIC    #177, 4(SP) ;; GET RID OF JUNK IF ANY
4109 014756 026627 000004 000023      CMP    4(SP), #23 ;; IS IT A CONTROL-S?
4110 014764 001013      BNE     3$      ;; BRANCH IF NO
4111 014766 105777 164152      2$: TSTB   $STKS    ;; WAIT FOR A CHARACTER
4112 014772 100375      BPL     2$      ;; LOOP UNTIL ITS THERE
4113 014774 117746 164146      MOVB   $STKB, -(SP) ;; GET CHARACTER
4114 015000 042716 177600      BIC    #177, (SP) ;; MAKE IT 7-BIT ASCII
4115 015004 022627 000021      CMP    (SP)+, #21 ;; IS IT A CONTROL-Q?
4116 015010 001366      BNE     2$      ;; IF NOT DISCARD IT
4117 015012 000750      BR      1$      ;; YES, RESUME
4118 015014 026627 000004 000140      3$: CMP    4(SP), #140 ;; IS IT UPPER CASE?
4119 015022 002407      BLT     4$      ;; BRANCH IF YES
4120 015024 026627 000004 000175      CMP    4(SP), #175 ;; IS IT A SPECIAL CHAR?
4121 015032 003003      BGT     4$      ;; BRANCH IF YES
4122 015034 042766 000040 000004      BIC    #40, 4(SP) ;; MAKE IT UPPER CASE
4123 015042 000002      4$: RTI      ;; GO BACK TO USER
4124      ;*****
4125      ;THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4126      ;CALL:
4127      ; RDLIN     ;; INPUT A STRING FROM THE TTY
4128      ; RETURN HERE ;; ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4129      ;
4130      ;
4131      ;
4131 015044 010346      $RDLIN: MOV      R3, -(SP) ;; SAVE R3
4132 015046 012703 015152      1$: MOV    $TTYIN, R3 ;; GET ADDRESS
4133 015052 022703 015162      2$: CMP    $TTYIN+8., R3 ;; BUFFER FULL?
4134 015056 101405      BLOS   4$      ;; BR IF YES
4135 015060 104410      RDCHR   ;; GO READ ONE CHARACTER FROM THE TTY

```

```

4136 015062 112613          MOV      (SP)+,(R3)      ;; GET CHARACTER
4137 015064 122713 000177    10$:    CMPB     #177,(R3)      ;; IS IT A RUBOUT
4138 015070 001003          BNE      3$             ;; SKIP IF NOT
4139 015072 104401 001166    4$:     TYPE     $QUES    ;; TYPE A '?'
4140 015076 000763          BR       1$             ;; CLEAR THE BUFFER AND LOOP
4141 015100 111337 015150    3$:     MOV      (R3),9$    ;; ECHO THE CHARACTER
4142 015104 104401 015150          TYPE     9$
4143 015110 122723 000015          CMPB     #15,(R3)+     ;; CHECK FOR RETURN
4144 015114 001356          BNE      2$             ;; LOOP IF NOT RETURN
4145 015116 105063 177777    CLR      -1(R3)        ;; CLEAR RETURN (THE 15)
4146 015122 104401 001170          TYPE     $LF           ;; TYPE A LINE FEED
4147 015126 012603          MOV      (SP)+,R3      ;; RESTORE R3
4148 015130 011646          MOV      (SP)-,(SP)    ;; ADJUST THE STACK AND PUT ADDRESS OF THE
4149 015132 016666 000004 000002    MOV      4(SP),2(SP)    ;; FIRST ASCII CHARACTER ON IT
4150 015140 012766 015152 000004    MOV      #STTYIN,4(SP)
4151 015146 000002          RTI
4152 015150          000          9$:     .BYTE     0           ;; RETURN
4153 015151          000          .BYTE     0           ;; STORAGE FOR ASCII CHAR. TO TYPE
4154 015152 000010          .BLKB     8           ;; TERMINATOR
4155 015162 052536 005015 000    $TTYIN:  .ASCIZ  /↑U/<15><12>  ;; RESERVE 8 BYTES FOR TTY INPUT
4156 015167          136 006507 000012    $CNTLG:  .ASCIZ  /↑G/<15><12>  ;; CONTROL "U"
4157 015174 005015 053523 020122    $MSWR:   .ASCIZ  <15><12>/SWR = /  ;; CONTROL "G"
4158 015202 020075          000
4159 015205          040 047040 053505    $MNEW:   .ASCIZ  / NEW = /
4160 015212 036440 000040
4161
4162          .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
4163
4164          ;; *****
4165          ;; *THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
4166          ;; *CHANGE IT TO BINARY.
4167          ;; *CALL:
4168          ;; *
4169          ;; *      RDOCT
4170          ;; *      RETURN HERE
4171          ;; *
4172          ;; *      READ AN OCTAL NUMBER
4173          ;; *      LOW ORDER BITS ARE ON TOP OF THE STACK
4174          ;; *      HIGH ORDER BITS ARE IN $HIOCT
4175
4176          SRDOCT: MOV      (SP)-,(SP)  ;; PROVIDE SPACE FOR THE
4177          MOV      4(SP),2(SP)  ;; INPUT NUMBER
4178          MOV      R0,-(SP)     ;; PUSH R0 ON STACK
4179          MOV      R1,-(SP)     ;; PUSH R1 ON STACK
4180          MOV      R2,-(SP)     ;; PUSH R2 ON STACK
4181          1$:    RDLIN          ;; READ AN ASCII LINE
4182          MOV      (SP)+,R0     ;; GET ADDRESS OF 1ST CHARACTER
4183          CLR      R1           ;; CLEAR DATA WORD
4184          CLR      R2
4185          2$:    MOV      (R0)+,-(SP)  ;; PICKUP THIS CHARACTER
4186          BEQ      3$           ;; IF ZERO GET OUT
4187          ASL      R1           ;; *2
4188          ROL      R2           ;; *4
4189          ASL      R1           ;; *8
4190          ROL      R2
4191          ASL      R1           ;; *8
4192          ROL      R2
4193          BIC      #↑C7,(SP)    ;; STRIP THE ASCII JUNK

```

```

4190 015270 062601          ADD      (SP)+,R1          ;; ADD IN THIS DIGIT
4191 015272 000764          BR       2$              ;; LOOP
4192 015274 005726          3$:    TST      (SP)+          ;; CLEAN TERMINATOR FROM STACK
4193 015276 010166          MOV      R1,12(SP)      ;; SAVE THE RESULT
4194 015302 010237          MOV      R2,$SHIOCT
4195 015306 012602          MOV      (SP)+,R2      ;; POP STACK INTO R2
4196 015310 012601          MOV      (SP)+,R1      ;; POP STACK INTO R1
4197 015312 012600          MOV      (SP)+,R0      ;; POP STACK INTO R0
4198 015314 000002          RTI
4199 015316 000000          $SHIOCT: .WORD 0        ;; RETURN
                                        .SBTTL TYPE ROUTINE      ;; HIGH ORDER BITS GO HERE

4200
4201
4202
4203 *****
4204 *ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
4205 *THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
4206 *NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
4207 *NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
4208 *NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
4209 *
4210 *CALL:
4211 *1) USING A TRAP INSTRUCTION
4212 *      TYPE ,MESADR          ;; MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
4213 *OR
4214 *      TYPE
4215 *      MESADR
4216 *
4217 015320 105737 001157 $TYPE: TSTB   $TPFLG          ;; IS THERE A TERMINAL?
4218 015324 100002          BPL     1$              ;; BR IF YES
4219 015326 000000          HALT
4220 015330 000430          BR     3$              ;; HALT HERE IF NO TERMINAL
4221 015332 010046          1$:    MOV     RO,-(SP)    ;; LEAVE
4222 015334 017600          MOV     R2,(SP),RO     ;; SAVE RO
4223 015340 122737 000001 001212 CMPB    #APTENV,$ENV    ;; GET ADDRESS OF ASCIZ STRING
4224 015346 001011          BNE    62$             ;; RUNNING IN APT MODE
4225 015350 132737 000100 001213 BITB    #APTPOOL,$ENVM  ;; NO GO CHECK FOR APT CONSOLE
4226 015356 001405          BEQ    62$             ;; SPOOL MESSAGE TO APT
4227 015360 010037 015370          MOV     RO,61$        ;; NO GO CHECK FOR CONSOLE
4228 015364 004737 015610          JSR    PC,$ATY3       ;; SETUP MESSAGE ADDRESS FOR APT
4229 015370 000000          .WORD 0               ;; SPOOL MESSAGE TO APT
4230 015372 132737 000040 001213 61$:    BITB    #APTCSUP,$ENVM  ;; MESSAGE ADDRESS
4231 015400 001003          BNE    60$             ;; APT CONSOLE SUPPRESSED
4232 015402 112046          MOV     (RO)+,-(SP)    ;; YES, SKIP TYPE OUT
4233 015404 001005          BNE    4$              ;; PUSH CHARACTER TO BE TYPED ONTO STACK
4234 015406 005726          TST     (SP)+          ;; BR IF IT ISN'T THE TERMINATOR
4235 015410 012600          60$:    MOV     (SP)+,RO   ;; IF TERMINATOR POP IT OFF THE STACK
4236 015412 062716 000002          3$:    ADD     #2,(SP)    ;; RESTORE RO
4237 015416 000002          RTI
4238 015420 122716 000011          4$:    CMPB    #HT,(SP)    ;; ADJUST RETURN PC
4239 015424 001430          BEQ    8$              ;; RETURN
4240 015426 122716 000200          CMPB    #CRLF,(SP)    ;; BRANCH IF <HT>
4241 015432 001006          BNE    5$              ;; BRANCH IF NOT <CRLF>
4242 015434 005726          TST     (SP)+          ;; POP <CR><LF> EQUIV
4243 015436 104401          TYPE

```

```

4244 015440 001167          $CRLF
4245 015442 105037 015576    CLRB   $CHARCNT      ;; CLEAR CHARACTER COUNT
4246 015446 000755          BR      2$           ;; GET NEXT CHARACTER
4247 015450 004737 015532    5$: JSR   PC,$TYPEC  ;; GO TYPE THIS CHARACTER
4248 015454 123726 001156    6$: CMPB  $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
4249 015460 001350          BNE    2$           ;; IF NO GO GET NEXT CHAR.
4250 015462 013746 001154    MOV    $NULL,-(SP)  ;; GET # OF FILLER CHARS. NEEDED
4251                                     AND THE NULL CHAR.
4252 015466 105366 000001    7$: DECB  1(SP)      ;; DOES A NULL NEED TO BE TYPED?
4253 015472 002770          BLT    6$           ;; BR IF NO--GO POP THE NULL OFF OF STACK
4254 015474 004737 015532    JSR   PC,$TYPEC  ;; GO TYPE A NULL
4255 015500 105337 015576    DECB  $CHARCNT     ;; DO NOT COUNT AS A COUNT
4256 015504 000770          BR      7$         ;; LOOP
4257
4258                                     ;HORIZONTAL TAB PROCESSOR
4259
4260 015506 112716 000040    8$: MOVB  #'(SP)     ;; REPLACE TAB WITH SPACE
4261 015512 004737 015532    9$: JSR   PC,$TYPEC  ;; TYPE A SPACE
4262 015516 132737 000007 015576    BITB  #7,$CHARCNT  ;; BRANCH IF NOT AT
4263 015524 001372          BNE    9$          ;; TAB STOP
4264 015526 005726          TST   (SP)+       ;; POP SPACE OFF STACK
4265 015530 000724          BR      2$         ;; GET NEXT CHARACTER
4266 015532 105777 163412    $TYPEC: TSTB  @STPS   ;; WAIT UNTIL PRINTER IS READY
4267 015536 100375          BPL   $TYPEC
4268 015540 116677 000002 163404    MOVB  2(SP),@STPB  ;; LOAD CHAR TO BE TYPED INTO DATA REG.
4269 015546 122766 000015 000002    CMPB  #CR,2(SP)   ;; IS CHARACTER A CARRIAGE RETURN?
4270 015554 001003          BNE    1$         ;; BRANCH IF NO
4271 015556 105037 015576    CLRB  $CHARCNT    ;; YES--CLEAR CHARACTER COUNT
4272 015562 000406          BR      $TYPEX    ;; EXIT
4273 015564 122766 000012 000002    1$: CMPB  #LF,2(SP)  ;; IS CHARACTER A LINE FEED?
4274 015572 001402          BEQ   $TYPEX     ;; BRANCH IF YES
4275 015574 105227          INCB (PC)+       ;; COUNT THE CHARACTER
4276 015576 000000          $CHARCNT: .WORD  0 ;; CHARACTER COUNT STORAGE
4277 015600 000207          $TYPEX: RTS      PC
4278
4279                                     .SBTTL  APT COMMUNICATIONS ROUTINE
4280
4281                                     ;*****
4282 015602 112737 000001 016046    $ATY1: MOVB  #1,$FFLG  ;; TO REPORT FATAL ERROR
4283 015610 112737 000001 016044    $ATY3: MOVB  #1,$MFLG  ;; TO TYPE A MESSAGE
4284 015616 000403          BR      $ATYC
4285 015620 112737 000001 016046    $ATY4: MOVB  #1,$FFLG  ;; TO ONLY REPORT FATAL ERROR
4286 015626          $ATYC:
4287 015626 010046          MOV   R0,-(SP)    ;; PUSH R0 ON STACK
4288 015630 010146          MOV   R1,-(SP)    ;; PUSH R1 ON STACK
4289 015632 105737 016044    TSTB  $MFLG       ;; SHOULD TYPE A MESSAGE?
4290 015636 001450          BEQ   5$          ;; IF NOT: BR
4291 015640 122737 000001 001212    CMPB  #APTENV,$ENV  ;; OPERATING UNDER APT?
4292 015646 001031          BNE   3$          ;; IF NOT: BR
4293 015650 132737 000100 001213    BITB  #APTPOOL,$ENVM  ;; SHOULD SPOOL MESSAGES?
4294 015656 001425          BEQ   3$          ;; IF NOT: BR
4295 015660 017600 000004          MOV   @4(SP),R0   ;; GET MESSAGE ADDR.
4296 015664 062766 000002 000004    ADD   #2,4(SP)    ;; BUMP RETURN ADDR.
4297 015672 005737 001172    1$: TST   $MSGTYPE  ;; SEE IF DONE W/ LAST XMISSION?

```

```

4298 015676 001375          BNE      1$          ;; IF NOT: WAIT
4299 015700 010037 001206   MOV      RD,$MSGAD  ;; PUT ADDR IN MAILBOX
4300 015704 105720          TSTB    (RD)+      ;; FIND END OF MESSAGE
4301 015706 001376          BNE      2$
4302 015710 163700 001206   SUB      $MSGAD,RD  ;; SUB START OF MESSAGE
4303 015714 006200          ASR      RD         ;; GET MESSAGE LNTH IN WORDS
4304 015716 010037 001210   MOV      RD,$MSGLGT ;; PUT LENGTH IN MAILBOX
4305 015722 012737 000004 001172   MOV      #4,$MSGTYPE ;; TELL APT TO TAKE MSG.
4306 015730 000413          BR       5$
4307 015732 017637 000004 015756 3$:   MOV      @4(SP),4$   ;; PUT MSG ADDR IN JSR LINKAGE
4308 015740 062766 000002 000004   ADD     #2,4(SP)    ;; BUMP RETURN ADDRESS
4309 015746 013746 177776   MOV     177776,-(SP) ;; PUSH 177776 ON STACK
4310 015752 004737 015320   JSR     PC,$TYPE    ;; CALL TYPE MACRO
4311 015756 000000          .WORD   0
4312 015760          5$:
4313 015760 105737 016046   10$:   TSTB    $FFLG      ;; SHOULD REPORT FATAL ERROR?
4314 015764 001416          BEQ     12$        ;; IF NOT: BR
4315 015766 005737 001212   TST     $ENV       ;; RUNNING UNDER APT?
4316 015772 001413          BEQ     12$
4317 015774 005737 001172   11$:   TST     $MSGTYPE   ;; FINISHED LAST MESSAGE?
4318 016000 001375          BNE     11$        ;; IF NOT: WAIT
4319 016002 017637 000004 001174   MOV     @4(SP),$FATAL ;; GET ERROR #
4320 016010 062766 000002 000004   ADD     #2,4(SP)    ;; BUMP RETURN ADDR.
4321 016016 005237 001172   INC     $MSGTYPE   ;; TELL APT TO TAKE ERROR
4322 016022 105037 016046   12$:   CLRB   $FFLG      ;; CLEAR FATAL FLAG
4323 016026 105037 016045   CLRB   $LFLG      ;; CLEAR LOG FLAG
4324 016032 105037 016044   CLRB   $MFLG      ;; CLEAR MESSAGE FLAG
4325 016036 012601          MOV     (SP)+,R1   ;; POP STACK INTO R1
4326 016040 012600          MOV     (SP)+,R0   ;; POP STACK INTO R0
4327 016042 000207          RTS     PC         ;; RETURN
4328 016044          000          $MFLG: .BYTE 0   ;; MESSG. FLAG
4329 016045          000          $LFLG: .BYTE 0   ;; LOG FLAG
4330 016046          000          $FFLG: .BYTE 0   ;; FATAL FLAG
4331          016050          .EVEN
4332          000200   APTSIZE=200
4333          000001   APTENV=001
4334          000100   APTSPool=100
4335          000040   APTCSUP=040
4336          .SBTTL POWER DOWN AND UP ROUTINES
4337
4338 ;; *****
4339 :POWER DOWN ROUTINE
4340 016050 012737 016210 000024 $PWRDN: MOV     #SILLUP,@#PWRVEC ;; SET FOR FAST UP
4341 016056 012737 000340 000026   MOV     #340,@#PWRVEC+2 ;; PRIO:7
4342 016064 010046          MOV     R0,-(SP)   ;; PUSH R0 ON STACK
4343 016066 010146          MOV     R1,-(SP)   ;; PUSH R1 ON STACK
4344 016070 010246          MOV     R2,-(SP)   ;; PUSH R2 ON STACK
4345 016072 010346          MOV     R3,-(SP)   ;; PUSH R3 ON STACK
4346 016074 010446          MOV     R4,-(SP)   ;; PUSH R4 ON STACK
4347 016076 010546          MOV     R5,-(SP)   ;; PUSH R5 ON STACK
4348 016100 017746 163034   MOV     @SWR,-(SP)  ;; PUSH @SWR ON STACK
4349 016104 010637 016214          MOV     SP,$$AVR6  ;; SAVE SP
4350 016110 012737 016122 000024   MOV     #SPWRUP,@#PWRVEC ;; SET UP VECTOR
4351 016116 000000          HALT

```



```

4352 016120 000776 BR -2 ;;HANG UP
4353
4354 ::*****
4355 :POWER UP ROUTINE
4356 016122 012737 016210 000024 $PWRUP: MOV $SILLUP,2#PWRVEC ;;SET FOR FAST DOWN
4357 016130 013706 016214 MOV $SAVR6,SP ;;GET SP
4358 016134 005037 016214 CLR $SAVR6 ;;WAIT LOOP FOR THE TTY
4359 016140 005237 016214 1$: INC $SAVR6 ;;WAIT FOR THE INC
4360 016144 001375 BNE 1$ ;;OF WORD
4361 016146 012677 162766 MOV (SP)+,2SWR ;;POP STACK INTO 2SWR
4362 016152 012605 MOV (SP)+,R5 ;;POP STACK INTO R5
4363 016154 012604 MOV (SP)+,R4 ;;POP STACK INTO R4
4364 016156 012603 MOV (SP)+,R3 ;;POP STACK INTO R3
4365 016160 012602 MOV (SP)+,R2 ;;POP STACK INTO R2
4366 016162 012601 MOV (SP)+,R1 ;;POP STACK INTO R1
4367 016164 012600 MOV (SP)+,R0 ;;POP STACK INTO R0
4368 016166 012737 016050 000024 MOV $SPWRDN,2#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4369 016174 012737 000340 000026 MOV #340,2#PWRVEC+2 ;;PRIO:7
4370 016202 104401 TYPE REPORT THE POWER FAILURE
4371 016204 016216 $PWRMG: .WORD $POWER ;;POWER FAIL MESSAGE POINTER
4372 016206 000002 RTI
4373 016210 000000 $SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED
4374 016212 000776 BR -2 ;;BEFORE THE POWER DOWN WAS COMPLETE
4375 016214 000000 $SAVR6: 0 ;;PUT THE SP HERE
4376 016216 005015 047520 042527 $POWER: .ASCIZ <15><12>"POWER"
4377 016224 000122
4378 .EVEN
4379 .SBTTL TRAP DEC0DER
4380
4381 ::*****
4382 :*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
4383 :*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
4384 :*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
4385 :*GO TO THAT ROUTINE.
4386
4387 016226 010046 $TRAP: MOV R0,-(SP) ;;SAVE R0
4388 016230 016600 000002 MOV 2(SP),R0 ;;GET TRAP ADDRESS
4389 016234 005740 TST -(R0) ;;BACKUP BY 2
4390 016236 111000 MOVB (R0),R0 ;;GET RIGHT BYTE OF TRAP
4391 016240 006300 ASL R0 ;;POSITION FOR INDEXING
4392 016242 016000 016262 MOV STRPAD(R0),R0 ;;INDEX TO TABLE
4393 016246 000200 RTS R0 ;;GO TO ROUTINE
4394
4395
4396 ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
4397
4398 016250 011646 $TRAP2: MOV (SP),-(SP) ;;MOVE THE PC DOWN
4399 016252 016666 000004 000002 MOV 4(SP),2(SP) ;;MOVE THE PSW DOWN
4400 016260 000002 RTI ;;RESTORE THE PSW
4401
4402 .SBTTL TRAP TABLE
4403
4404 :*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
4405 :*BY THE "TRAP" INSTRUCTION.
    
```

Address	Word	Byte 1	Byte 2	Byte 3	Byte 4	Label	Text
4406							
4407							
4408							
4409	016262	016250					
4410	016264	015320					
4411	016266	013300					
4412	016270	013254					
4413	016272	013314					
4414	016274	013502					
4415							
4416	016276	014512					
4417							
4418	016300	014442					
4419	016302	014724					
4420	016304	015044					
4421	016306	015216					
4422	016310	020200	020040	020040			
4423	016316	047516	042524	072			
4424	016323	200	047506	020122			
4425	016330	044124	051511	050040			
4426	016336	047522	051107	046501			
4427	016344	052040	020117	052522			
4428	016352	020116	051120	050117			
4429	016360	051105	054514	020054			
4430	016366	053523	052111	044103			
4431	016374	033600	020054	043117			
4432	016402	052040	042510	053040			
4433	016410	041505	047524	020122			
4434	016416	042101	051104	051505			
4435	016424	020123	053523	052111			
4436	016432	044103	050040	041501			
4437	016440	020113	042450	033067			
4438	016446	026051					
4439	016450	046600	051525	020124			
4440	016456	042502	047440	027116			
4441	016464	024040	034115	030062			
4442	016472	026460	041531	041040			
4443	016500	040517	042122	100051			
4444	016506	000					
4445							
4446	016507	200	041125	051123	EM1:	.ASCIZ	<200>#UBSR ERROR#
4447	016514	042440	051122	051117			
4448	016522	000					
4449	016523	200	047511	051123	EM2:	.ASCIZ	<200>#IOSR ERROR#
4450	016530	042440	051122	051117			
4451	016536	000					
4452	016537	200	044523	042504	EM3:	.ASCIZ	<200>#SIDE 1 (IO TO UB) FAST PATH DATA ERROR#
4453	016544	030440	024040	047511			
4454	016552	052040	020117	041125			
4455	016560	020051	040506	052123			
4456	016566	050040	052101	020110			
4457	016574	040504	040524	042440			
4458	016602	051122	051117	000			
4459	016607	200	044523	042504	EM4:	.ASCIZ	<200>#SIDE 2 (UB TO IO) FAST PATH DATA ERROR#

```

; ROUTINE
;-----
$TRPAD: .WORD $TRAP2
        $TYPE  ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
        $TYPOC ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
        $TYPOS ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
        $TYPON ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
        $TYPDS ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

$GTSWR  ;;CALL=GTSWR          TRAP+6(104406)  GET SOFT-SWR SETTING

$CKSWR  ;;CALL=CKSWR          TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
$RDCHR  ;;CALL=RDCHR          TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
$RDLIN  ;;CALL=RDLIN          TRAP+11(104411) TTY TYPEIN STRING ROUTINE
$RDOCT  ;;CALL=RDOCT          TRAP+12(104412) READ AN OCTAL NUMBER FROM TTY

```

```

MESWCH: .ASCII <200># NOTE:#
        .ASCII <200>#FOR THIS PROGRAM TO RUN PROPERLY, SWITCH#

```

```

.ASCII <200>#7, OF THE VECTOR ADDRESS SWITCH PACK (E76),#

```

```

.ASCIIZ <200>#MUST BE ON. (M8200-YC BOARD)#<200>

```

4460	016614	031040	024040	041125	
4461	016622	052040	020117	047511	
4462	016630	020051	040506	052123	
4463	016636	050040	052101	020110	
4464	016644	040504	040524	042440	
4465	016652	051122	051117	000	
4466	016657	200	044523	042504	EM5: .ASCIZ <200>#SIDE 1 (IO TO UB) SILO DATA ERROR#
4467	016664	030440	024040	047511	
4468	016672	052040	020117	041125	
4469	016700	020051	044523	047514	
4470	016706	042040	052101	020101	
4471	016714	051105	047522	000122	
4472	016722	051600	042111	020105	EM6: .ASCIZ <200>#SIDE 2 (UB TO IO) SILO DATA ERROR#
4473	016730	020062	052450	020102	
4474	016736	047524	044440	024517	
4475	016744	051440	046111	020117	
4476	016752	040504	040524	042440	
4477	016760	051122	051117	000	
4478	016765	200	044523	042504	EM7: .ASCIZ <200>#SIDE 1 INPO DATA ERROR#
4479	016772	030440	044440	050116	
4480	017000	020060	040504	040524	
4481	017006	042440	051122	051117	
4482	017014	000			
4483	017015	200	044523	042504	EM10: .ASCIZ <200>#SIDE 2 INPO DATA ERROR#
4484	017022	031040	044440	050116	
4485	017030	020060	040504	040524	
4486	017036	042440	051122	051117	
4487	017044	000			
4488	017045	200	044515	051103	EM11: .ASCIZ <200>#MICRO-P DETECTED IPBM ERROR#
4489	017052	026517	020120	042504	
4490	017060	042524	052103	042105	
4491	017066	044440	041120	020115	
4492	017074	051105	047522	000122	
4493					
4494	017102	042600	051122	041520	DH1: .ASCIZ <200>#ERRPC GDDAT BDDAT KMCADR#
4495	017110	020040	043440	042104	
4496	017116	052101	020040	041040	
4497	017124	042104	052101	020040	
4498	017132	045440	041515	042101	
4499	017140	000122			
4500	017142	042600	051122	041520	DH7: .ASCIZ <200>#ERRPC INPO # GDDAT BDDAT KMCADR#
4501	017150	020040	044440	050116	
4502	017156	020060	020043	043440	
4503	017164	042104	052101	020040	
4504	017172	041040	042104	052101	
4505	017200	020040	045440	041515	
4506	017206	042101	000122		
4507	017212	052200	051505	020124	DH11: .ASCIZ <200>#TEST ERRPC KMCADR CSR#
4508	017220	020040	042440	051122	
4509	017226	041520	020040	045440	
4510	017234	041515	042101	020122	
4511	017242	041440	051123	000	
4512					
4513	017250				.EVEN

4514								
4515	017250	001116	001124	001126	DT1:	.WORD	\$ERRPC,\$GDDAT,\$BDDAT,KMCSR,0	
4516	017256	001424	000000					
4517	017262	001116	001460	001124	DT7:	.WORD	\$ERRPC,INNU,\$GDDAT,\$BDDAT,KMCSR,0	
4518	017270	001126	001424	000000				
4519	017276	001102	001116	001462	DT11:	.WORD	\$STS INM,\$ERRPC,KLAD,\$BDDAT,0	
4520	017304	001126	000000					
4521								
4522	017310	000000	000000		DF0:	0,0		
4523								
4524	017314	017322			SENA:	.WORD	SENBUF	
4525	017316	017722			RECB:	.WORD	RECBUF	
4526	017320	020122			RE2B:	.WORD	RE2BUF	
4527	017322	000200			SENBUF:	.BLKW	128.	
4528	017722	000100			RECBUF:	.BLKW	64.	
4529	020122	000100			RE2BUF:	.BLKW	64.	
4530		000001			.END			

Variable	Value	236	277	373	374	375	376	377	378	379	380	381	382	
ABASE =	170460	152#	236	277	373	374	375	376	377	378	379	380	381	382
ACDW1 =	000000	383	389	390	391	392	393	394	395	396	397	398	399	
ACDW2 =	000000	236	279											
ACPUOP =	000000	236	251											
ADDW0 =	000000	236												
ADDW1 =	000000	236												
ADDW10 =	000000	236												
ADDW11 =	000000	236												
ADDW12 =	000000	236												
ADDW13 =	000000	236												
ADDW14 =	000000	236												
ADDW15 =	000000	236												
ADDW2 =	000000	236												
ADDW3 =	000000	236												
ADDW4 =	000000	236												
ADDW5 =	000000	236												
ADDW6 =	000000	236												
ADDW7 =	000000	236												
ADDW8 =	000000	236												
ADDW9 =	000000	236												
ADEVCT =	000000	236	242											
ADEVM =	000000	236	278											
AENV =	000000	236	247											
AENVM =	000000	236	248											
AFATAL =	000000	236	239											
AMADR1 =	000000	236	264											
AMADR2 =	000000	236	268											
AMADR3 =	000000	236	271											
AMADR4 =	000000	236	274											
AMAMS1 =	000000	236	258											
AMAMS2 =	000000	236	266											
AMAMS3 =	000000	236	269											
AMAMS4 =	000000	236	272											
AMSGAD =	000000	236	244											
AMSLG =	000000	236	245											
AMSGTY =	000000	236	238											
AMTYP1 =	000000	236	259											
AMTYP2 =	000000	236	267											
AMTYP3 =	000000	236	270											
AMTYP4 =	000000	236	273											
APASS =	000000	236	241											
APRIOR =	000300	154#	236											
APTCSU =	000040	4230	4335#											
APTENV =	000001	3906	4223	4291	4333#									
APTSIZ =	000200	444	4332#											
APTSP0 =	000100	4225	4293	4334#										
ASWREG =	000000	236	249											
AESTN =	000000	236	240											
AUNIT =	000000	236	243											
AUSWR =	000000	236	250											
AVECT1 =	000300	153#	236	275										
AVECT2 =	000000	236	276											
BINT	012524	2062	2117	2801	2832	2861	2905	2932	2961	3014	3046	3049	3072	3118



DSWR = 177570	57#	217	431															
DT1 017250	302	309	316	323	330	337	4515#											
DT11 017276	358	4519#																
DT7 017262	344	351	4517#															
EMTVEC= 000030	146#	419*	420*															
EM1 016507	300	4446#																
EM10 017015	349	4483#																
EM11 017045	356	4488#																
EM2 016523	307	4449#																
EM3 016537	314	4452#																
EM4 016607	321	4459#																
EM5 016657	328	4466#																
EM6 016722	335	4473#																
EM7 016765	342	4478#																
ERCNT 001456	403#	448*	3361	3918*														
ERRIS 012050	3424	3440#																
ERRVEC= 000004	139#	429	430*	441*	3989	3990*	3992*	3995*										
FILBUF 012542	3176	3252	3599#															
FILEX 012576	3605	3609#																
GNS = ***** U	30	497	3352	3359	3366	3431	3566	3570	3578	3626	3634	3641	3647					
	3657	3672	3686	3694	3700	3707	3726	4410	4411	4412	4413	4414	4416					
	4418	4419	4420	4421														
	492	4416#																
GTSWR = 104406	49#	4238	4279															
HT = 000011	404#	584*	621*	660*	697*	4517												
INNU 001460	144#	417*	418*															
IOTVEC= 000020	3416	3421	3437#	3441	3456													
ISSEX 012044	519	526	578	654	738	746	812	858	913	980	990	1016	1049					
ISSUEC 011656	1059	1085	1118	1128	1154	1187	1197	1223	1256	1266	1292	1325	1335					
	1361	1394	1404	1430	1463	1473	1499	1545	1571	1612	1638	1679	1705					
	1746	1772	1813	1839	1880	1906	1947	1973	2014	2040	2065	2146	2173					
	2206	2239	2272	2305	2338	2371	2404	2437	2476	2508	2540	2572	2604					
	2636	2668	2700	2732	2764	2773	2784	2806	2815	2842	2937	2976	2985					
	2996	3019	3026	3051	3179	3254	3403#											
KLAD 001462	405#	3442*	3548*	4519														
KMADR 001426	390#	3527*	3539*	3544*	3548	3552*												
KMCSR 001424	389#	451	457	464	520	527	579	655	739	747	813	859	914					
	981	991	1017	1050	1060	1086	1119	1129	1155	1188	1198	1224	1257					
	1267	1293	1326	1336	1362	1395	1405	1431	1464	1474	1500	1546	1572					
	1613	1639	1680	1706	1747	1773	1814	1840	1881	1907	1948	1974	2015					
	2041	2066	2147	2174	2207	2240	2273	2306	2339	2372	2405	2438	2477					
	2509	2541	2573	2605	2637	2669	2701	2733	2765	2774	2785	2807	2816					
	2843	2938	2977	2986	2997	3020	3027	3052	3180	3255	3324	3525*	3529*					
	3533*	3538*	3547*	3551*	3555*	3592*	4515	4517										
KMDBR 001430	391#	3531*	3549	3580														
KMMADR 001376	374#	466*	469*															
KMMCSR 001374	373#	464*	465*	466	467	471	3593*											
KMMDBR 001400	375#	467*	470*															
KMSIM 012100	550	616	692	729	771	791	833	867	873	934	996	1022	1065					
	1091	1134	1160	1203	1229	1272	1298	1341	1367	1410	1436	1479	1505					
	1531	1539	1565	1598	1606	1632	1665	1673	1699	1732	1740	1766	1799					
	1807	1833	1866	1874	1900	1933	1941	1967	2000	2008	2034	2071	2093					
	2119	2124	2179	2212	2245	2278	2311	2344	2377	2410	2443	2470	2502					
	2534	2566	2598	2630	2662	2694	2726	2836	2873	2880	2889	2909	2916					

M08

MD-11-DRLPN-A  
DRLPN.P11

MACY11 27(654) 15-DEC-77 08:43 PAGE 91  
CROSS REFERENCE TABLE

SEQ 0103

	2943	3085	3092	3101	3122	3127	3150	3191	3197	3205	3266	3273	3472*
KWADR 001422	385#												
KWINT 012236	3512#												
LF = 000012	50#	4273	4279										
LOAD 012246	504	3522#											
LOADER= 012402	3550	3560#											
LSTED 001724	451#	3326											
MBSLO 001402	376#	472											
MBSL1 001404	377#	3497*	3501*	3508*									
MBSL2 001406	378#												
MBSL3 001410	379#	624	700	796	865*	1002	1028	1071	1097	1140	1166	1209	1235
	1278	1304	1347	1373	1416	1442	1485	1511	1538*	1564*	1605*	1631*	1672*
	1698*	1739*	1765*	1806*	1832*	1873*	1899*	1940*	1966*	2007*	2033*	2098	2118*
	2185	2218	2251	2284	2317	2350	2383	2416	2449	2469*	2501*	2533*	2565*
	2597*	2629*	2661*	2693*	2725*	3196	3202	3265*					
MBSL4 001412	380#												
MBSL5 001414	381#	556	776	838	879	939	2076	2878	2894	2895	2921	2922	2948
	2950	3090	3106	3108	3133	3134	3136	3157	3211				
MBSL6 001416	382#	3499*											
MBSL7 001420	383#												
MCNT 001366	364#	3321											
MESWCH 016310	503	4422#											
MOCRAM 012632	37	3623#											
MRCODE= *****	1#	3477	3482	3523	3545								
NCNT 001372	366#	450*	3321	3323*									
ODT 013070	39	3683#	3714										
PC =%000007	70#	2062*	2117*	2801*	2832*	2861*	2905*	2932*	2961*	3014*	3046*	3049*	3072*
	3118*	3146*	3149*	3170*	3176*	3226*	3252*	3294*	3342*	3345*	3372*	3377	3434*
	3487*	3509*	3594*	3602*	3609*	3615*	3903*	3909*	3958*	4076*	4228*	4247*	4254*
	4261*	4275*	4277*	4310*	4327*								
	56#												
PIRQ = 177772	150#												
PIRQVE= 000240	73#												
PRO = 000000	74#												
PR1 = 000040	75#												
PR2 = 000100	76#												
PR3 = 000140	77#												
PR4 = 000200	78#												
PR5 = 000240	79#												
PR6 = 000300	80#												
PR7 = 000340	53#	54											
PS = 177776	54#												
PSW = 177776	145#	423*	424*	4340*	4341*	4350*	4356*	4368*	4369*				
PWRVEC= 000024	3603	3611	3612*	3613*	3617#								
RANA 012626	3606	3611*	3613	3614*	3618#								
RANB 012630	3602	3611#											
RAND 012600	3427#												
RBST = 011770	4135	4419#											
RDCHR = 104410	4177	4420#											
RDLIN = 104411	3628	3636	3674	3688	3728	4421#							
RDOCT = 104412	4525#												
RECB 017316	3188	3228	3260	3296	3604	3607	4525	4528#					
RECBUF 017722	140#												
RESVEC= 000010	4526#												
RE28 017320													

G

























.SWRLO	24#	
.TRMTR	1#	
.SACT1	1#	157
.SAPT8	233#	
.SAPTH	1#	168
.SAPTY	1#	4279
.SCATC	1#	24
.SCMTA	1#	190
.SEOP	1#	3329
.SERRO	1#	3876
.SERRT	1#	3921
.SPOWE	1#	4336
.SRDOC	1#	4162
.SREAD	1#	4022
.SSCOP	1#	3968
.STRAP	1#	4379
.STYPD	1#	3809
.STYBE	1#	4200
.STYPO	1#	3732

ADC	3612	3614													
ADD	453	455	460	465	469	470	475	3325	3413	3477	3482	3611	3613	3717	3760
ASL	3770	3841	3943	4061	4070	4190	4236	4296	4308	4320					
ASLB	3476	3480	3940	3941	3942	4084	4085	4086	4183	4185	4187	4391			
ASR	3846														
BCC	4303														
BEQ	3847														
	445	489	538	561	589	626	665	702	758	780	821	842	883	899	942
	957	1004	1029	1073	1098	1142	1167	1211	1236	1280	1305	1349	1374	1418	1443
	1487	1512	1554	1579	1621	1646	1688	1713	1755	1780	1822	1847	1889	1914	1956
	1981	2023	2048	2079	2132	2187	2220	2253	2286	2319	2352	2385	2418	2451	2485
	2517	2549	2581	2613	2645	2677	2709	2741	2792	2851	2896	2951	3003	3061	3107
	3159	3214	3232	3282	3300	3322	3370	3441	3605	3668	3676	3714	3787	3892	3895
	3919	3945	3950	3963	3999	4001	4003	4009	4041	4068	4083	4182	4226	4239	4274
	4290	4294	4314	4316											
BGE	4012														
BGT	3344	3794	3855	4080	4121										
BIC	2849	2949	3057	3155	3341	3475	3481	3662	3712	3784	4037	4054	4081	4108	4114
BICB	4122	4189													
BIS	3410	3508													
BISB	2813	2915	3032	3058	3132	3156	3529	3533	3547	3630	3651	3678	3789	3790	3849
BIT	3850	4088													
BITB	3412	3501	3932												
	3490	3894	3901	3984	3998	4006									
	444	2770	2791	2822	2850	2878	2895	2922	2950	2982	3002	3034	3090	3106	3134
	4225	4230	4262	4293											
BLOS	4134														
BLT	3795	3838	3854	4078	4119	4253									
BMI	3424	3486	3845												
BNE	414	434	462	477	483	487	491	500	502	797	920	2099	2153	2771	2780
	2823	2879	2886	2923	2983	2992	3035	3091	3098	3135	3204	3246	3272	3314	3416
	3421	3426	3491	3541	3550	3554	3608	3664	3716	3720	3785	3843	3902	3907	3933
	3955	3985	4007	4033	4039	4059	4066	4073	4110	4116	4138	4144	4224	4231	4233
	4241	4249	4263	4270	4292	4298	4301	4318	4360						
BPL	3406	3562	3660	3710	3783	3829	3859	3914	4035	4051	4106	4112	4218	4267	
BR	436	493	496	768	789	808	830	893	908	931	951	2090	2143	2802	2833
	2906	2933	3015	3047	3119	3147	3227	3244	3295	3312	3351	3358	3365	3417	3430
	3456	3489	3565	3569	3577	3625	3633	3640	3646	3656	3666	3671	3679	3685	3693
	3699	3706	3718	3722	3725	3730	3761	3776	3797	3840	3857	3912	3938	3965	3987
	3993	3996	4062	4089	4091	4117	4140	4191	4220	4246	4256	4265	4272	4284	4306
	4352	4374													
CLR	412	425	443	448	586	623	662	699	1013	1082	1151	1220	1289	1358	1427
	1496	1563	1630	1697	1764	1831	1898	1965	2032	3210	3278	3338	3339	3478	3522
	3525	3527	3538	3544	3551	3593	3649	3774	3832	3835	3931	4005	4048	4049	4179
	4180	4358													
CLRB	534	557	754	1014	1083	1152	1221	1290	1359	1428	1497	1564	1631	1698	1765
CMP	1832	1899	1966	2033	3444	3861	4004	4145	4245	4271	4322	4323	4324		
	413	433	490	537	560	757	779	820	841	882	898	941	956	1003	1072
	1141	1210	1279	1348	1417	1486	1553	1620	1687	1754	1821	1888	1955	2022	2078
	2131	3321	3540	3549	3553	3604	3607	3853	3994	4011	4032	4038	4058	4065	4077
	4079	4109	4115	4118	4120	4133									
CMPB	488	588	625	664	701	2186	2219	2252	2285	2318	2351	2384	2417	2450	2484
	2516	2548	2580	2612	2644	2676	2708	2740	3060	3158	3213	3231	3281	3299	3440
	3663	3667	3713	3715	3719	3906	4000	4040	4072	4137	4143	4223	4238	4240	4248

	4269	4273	4291	2885	2991	3097	3203	3245	3271	3313	3342	3939			
DEC	461	476	2779	2885											
DECb	3782	3793	4252	4255											
EMT	45														
HALT	30	3582	3915	4219	4351	4373									
INC	482	499	501	3323	3340	3425	3512	3539	3552	3665	3788	3796	3839	3897	3918
INCB	4010	4087	4321	4359											
IOT	3561	3891	4015	4275											
JMP	46														
JSR	35	37	39	3326	3377	3435	3584								
	504	519	526	550	578	616	654	692	729	738	746	771	791	812	833
	858	867	873	913	934	980	990	996	1016	1022	1049	1059	1065	1085	1091
	1118	1128	1134	1154	1160	1187	1197	1203	1223	1229	1256	1266	1272	1292	1298
	1325	1335	1341	1361	1367	1394	1404	1410	1430	1436	1463	1473	1479	1499	1505
	1531	1539	1545	1565	1571	1598	1606	1612	1632	1638	1665	1673	1679	1699	1705
	1732	1740	1746	1766	1772	1799	1807	1813	1833	1839	1866	1874	1880	1900	1906
	1933	1941	1947	1967	1973	2000	2008	2014	2034	2040	2062	2065	2071	2093	2117
	2119	2124	2146	2173	2179	2206	2212	2239	2245	2272	2278	2305	2311	2338	2344
	2371	2377	2404	2410	2437	2443	2470	2476	2502	2508	2534	2540	2566	2572	2598
	2604	2630	2636	2662	2668	2694	2700	2726	2732	2764	2773	2784	2801	2806	2815
	2832	2836	2842	2861	2873	2880	2889	2905	2909	2916	2932	2937	2943	2961	2976
	2985	2996	3014	3019	3026	3046	3049	3051	3072	3085	3092	3101	3118	3122	3127
	3146	3149	3150	3170	3176	3179	3191	3197	3205	3226	3252	3254	3266	3273	3294
	3372	3434	3487	3602	3603	3909	4076	4228	4247	4254	4261	4310			
MOV	411	415	417	418	419	420	421	422	423	424	426	429	430	431	432
	437	439	440	441	446	449	450	451	452	454	456	457	458	459	464
	466	467	471	472	473	474	510	511	514	515	535	548	558	584	585
	621	622	660	661	697	698	755	777	798	819	839	880	896	921	940
	954	987	1056	1125	1194	1263	1332	1401	1470	1537	1604	1671	1738	1805	1872
	1939	2006	2077	2100	2130	2155	2170	2203	2236	2269	2302	2335	2368	2401	2434
	2468	2500	2532	2564	2596	2628	2660	2692	2724	2757	2790	2822	2854	2888	2920
	2968	2972	2993	3078	3082	3099	3188	3189	3212	3228	3229	3230	3263	3264	3280
	3296	3297	3298	3324	3345	3354	3361	3369	3403	3404	3414	3433	3437	3442	3472
	3473	3474	3485	3493	3494	3499	3523	3531	3545	3548	3555	3574	3580	3592	3599
	3603	3606	3629	3637	3643	3650	3652	3653	3675	3677	3690	3696	3702	3729	3757
	3765	3766	3767	3773	3780	3798	3799	3800	3801	3802	3822	3823	3824	3825	3826
	3827	3828	3833	3836	3856	3862	3863	3864	3865	3866	3868	3869	3893	3898	3930
	3935	3944	3949	3954	3956	3960	3989	3990	3992	3995	4013	4014	4017	4018	4019
	4045	4069	4074	4103	4104	4131	4132	4147	4148	4149	4150	4172	4173	4174	4175
	4176	4178	4193	4194	4195	4196	4197	4221	4222	4227	4235	4250	4287	4288	4295
	4299	4304	4305	4307	4309	4319	4325	4326	4340	4341	4342	4343	4344	4345	4346
	4347	4348	4349	4350	4356	4357	4361	4362	4363	4364	4365	4366	4367	4368	4369
MOVB	4387	4388	4392	4398	4399										
	494	513	533	556	587	624	663	700	735	753	776	796	818	838	865
	879	895	919	939	953	988	1002	1028	1057	1071	1097	1126	1140	1166	1195
	1209	1235	1264	1278	1304	1333	1347	1373	1402	1416	1442	1471	1485	1511	1538
	1552	1578	1605	1619	1645	1672	1686	1712	1739	1753	1779	1806	1820	1846	1873
	1887	1913	1940	1954	1980	2007	2021	2047	2063	2076	2098	2118	2129	2152	2171
	2185	2204	2218	2237	2251	2270	2284	2303	2317	2336	2350	2369	2383	2402	2416
	2435	2449	2469	2483	2501	2515	2533	2547	2565	2579	2597	2611	2629	2643	2661
	2675	2693	2707	2725	2739	2790	2821	2848	2894	2921	2948	3004	3033	3036	3059
	3108	3133	3136	3157	3196	3202	3211	3233	3234	3265	3279	3301	3302	3407	3408
	3409	3411	3419	3443	3479	3497	3661	3711	3758	3759	3762	3763	3764	3768	3771
	3772	3791	3831	3834	3848	3851	3860	3900	3908	4016	4036	4053	4107	4113	4136

NEG	4141	4181	4232	4260	4268	4282	4283	4285	4390								
NOP	3769	3830															
RESET	509	3337	3373	3374	3375	3498	3500	3502	3503	3504	3505	3506	3507				
ROL	478	3371	3591														
RTI	3775	3777	3778	3779	3781	4184	4186	4188									
RTS	438	3513	3803	3870	3920	4020	4075	4123	4151	4198	4237	4372	4400				
SUB	3438	3495	3509	3556	3594	3609	3615	3958	4277	4327	4393						
TRAP	3721	3837	3899	4302													
TST	4402	4411	4412	4413	4414	4416	4418	4419	4420	4421							
TSTB	486	3572	3786	3842	3852	3913	3962	3991	4008	4067	4082	4192	4234	4242	4264		
	4297	4315	4317	4389													
	3405	3415	3420	3423	3659	3709	3844	3858	4002	4034	4050	4105	4111	4217	4266		
.ASCII	4289	4300	4313														
.ASCIZ	229	230	4422	4424	4431												
	228	231	498	3353	3360	3367	3432	3567	3571	3579	3627	3635	3642	3648	3658		
	3673	3687	3695	3701	3708	3727	3966	4155	4156	4157	4159	4376	4439	4446	4449		
	4452	4459	4466	4472	4478	4483	4488	4494	4500	4507							
.ASECT	1																
.BLKB	4154																
.BLKW	3875	4527	4528	4529													
.BYTE	199	200	205	206	214	215	223	224	225	226	247	248	258	259	266		
	267	269	270	272	273	3379	3804	3805	3806	3807	3910	3911	4152	4153	4328		
	4329	4330															
.DSABL	4092																
.ENABL	1	4025															
.END	4530																
.ENDC	7																
	197	21	23	24	33	45	137	151	160	164	166	171	173	180	193		
	279	199	227	228	229	233	236	258	266	269	272	275	276	277	278		
	498	282	407	415	416	419	421	423	425	426	427	448	484	490	496		
	531	507	508	509	510	511	512	520	521	522	523	524	527	528	530		
	579	539	545	546	547	548	549	551	553	554	562	572	573	574	575		
	648	580	581	582	583	590	611	612	613	614	617	618	619	620	627		
	694	649	650	651	655	656	657	658	659	666	687	688	689	690	693		
	742	695	696	703	724	725	726	727	730	731	732	733	739	740	741		
	809	743	747	748	750	751	769	772	774	775	790	792	793	794	795		
	860	813	814	816	817	831	834	836	837	843	853	854	855	856	859		
	916	861	862	863	868	869	870	871	874	876	877	894	909	914	915		
	982	917	918	932	935	937	938	952	958	970	971	976	977	978	981		
	1019	984	985	991	992	993	994	995	997	998	999	1000	1009	1017	1018		
	1051	1020	1021	1023	1024	1025	1026	1030	1034	1039	1040	1045	1046	1047	1050		
	1088	1053	1054	1060	1061	1062	1063	1064	1066	1067	1068	1069	1078	1086	1087		
	1120	1089	1090	1092	1093	1094	1095	1099	1103	1108	1109	1114	1115	1116	1119		
	1157	1122	1123	1129	1130	1131	1132	1133	1135	1136	1137	1138	1147	1155	1156		
	1189	1158	1159	1161	1162	1163	1164	1168	1172	1177	1178	1183	1184	1185	1188		
	1226	1191	1192	1198	1199	1200	1201	1202	1204	1205	1206	1207	1216	1224	1225		
	1258	1227	1228	1230	1231	1232	1233	1237	1241	1246	1247	1252	1253	1254	1257		
	1295	1260	1261	1267	1268	1269	1270	1271	1273	1274	1275	1276	1285	1293	1294		
	1327	1296	1297	1299	1300	1301	1302	1306	1310	1315	1316	1321	1322	1323	1326		
	1364	1329	1330	1336	1337	1338	1339	1340	1342	1343	1344	1345	1354	1362	1363		
	1396	1365	1366	1368	1369	1370	1371	1375	1379	1384	1385	1390	1391	1392	1395		
	1433	1398	1399	1405	1406	1407	1408	1409	1411	1412	1413	1414	1423	1431	1432		
	1465	1434	1435	1437	1438	1439	1440	1444	1448	1453	1454	1459	1460	1461	1464		
	1502	1467	1468	1474	1475	1476	1477	1478	1480	1481	1482	1483	1492	1500	1501		
		1503	1504	1506	1507	1508	1509	1513	1517	1522	1523	1528	1529	1530	1532		

1534	1535	1540	1541	1542	1543	1546	1547	1548	1549	1550	1559	1566	1567	1568
1569	1572	1573	1574	1575	1576	1580	1584	1589	1590	1595	1596	1597	1599	1601
1602	1607	1608	1609	1610	1613	1614	1615	1616	1617	1626	1633	1634	1635	1636
1639	1640	1641	1642	1643	1647	1651	1656	1657	1662	1663	1664	1666	1668	1669
1674	1675	1676	1677	1680	1681	1682	1683	1684	1693	1700	1701	1702	1703	1706
1707	1708	1709	1710	1714	1718	1723	1724	1729	1730	1731	1733	1735	1736	1741
1742	1743	1744	1747	1748	1749	1750	1751	1760	1767	1768	1769	1770	1773	1774
1775	1776	1777	1781	1785	1790	1791	1796	1797	1798	1800	1802	1803	1808	1809
1810	1811	1814	1815	1816	1817	1818	1827	1834	1835	1836	1837	1840	1841	1842
1843	1844	1848	1852	1857	1858	1863	1864	1865	1867	1869	1870	1875	1876	1877
1878	1881	1882	1883	1884	1885	1894	1901	1902	1903	1904	1907	1908	1909	1910
1911	1915	1919	1924	1925	1930	1931	1932	1934	1936	1937	1942	1943	1944	1945
1948	1949	1950	1951	1952	1961	1968	1969	1970	1971	1974	1975	1976	1977	1978
1982	1986	1991	1992	1997	1998	1999	2001	2003	2004	2009	2010	2011	2012	2015
2016	2017	2018	2019	2028	2035	2036	2037	2038	2041	2042	2043	2044	2045	2049
2053	2058	2059	2060	2061	2066	2067	2068	2069	2070	2072	2074	2075	2091	2094
2095	2096	2097	2100	2113	2114	2115	2116	2120	2121	2122	2123	2125	2127	2128
2144	2147	2148	2149	2150	2151	2154	2166	2167	2168	2169	2174	2175	2176	2177
2178	2180	2181	2182	2183	2188	2193	2199	2200	2201	2202	2207	2208	2209	2210
2211	2213	2214	2215	2216	2221	2226	2233	2233	2234	2235	2240	2241	2242	2243
2244	2246	2247	2248	2249	2254	2259	2266	2266	2267	2268	2273	2274	2275	2276
2277	2279	2280	2281	2282	2287	2299	2298	2299	2300	2301	2306	2307	2308	2309
2310	2312	2313	2314	2315	2320	2335	2331	2333	2333	2334	2336	2340	2341	2342
2343	2345	2346	2347	2348	2353	2358	2364	2364	2366	2367	2372	2373	2374	2375
2376	2378	2379	2380	2381	2386	2391	2397	2398	2399	2400	2405	2406	2407	2408
2409	2411	2412	2413	2414	2419	2424	2430	2431	2432	2433	2438	2439	2440	2441
2442	2444	2445	2446	2447	2452	2457	2464	2465	2466	2467	2471	2472	2473	2474
2477	2478	2479	2480	2481	2486	2490	2496	2498	2498	2499	2503	2504	2505	2506
2509	2510	2511	2512	2513	2518	2529	2536	2536	2537	2537	2550	2551	2553	2553
2541	2542	2543	2544	2545	2550	2554	2561	2562	2563	2563	2567	2568	2569	2570
2573	2574	2575	2576	2577	2582	2586	2593	2593	2595	2595	2600	2600	2601	2602
2605	2606	2607	2608	2609	2614	2618	2624	2625	2627	2627	2631	2632	2633	2634
2637	2638	2639	2640	2641	2646	2650	2656	2657	2658	2659	2663	2664	2665	2666
2669	2670	2671	2672	2673	2678	2680	2686	2689	2690	2691	2695	2696	2697	2698
2701	2702	2703	2704	2705	2710	2714	2720	2721	2722	2723	2728	2728	2729	2730
2733	2734	2735	2736	2737	2742	2746	2754	2754	2756	2757	2765	2765	2766	2768
2769	2774	2775	2776	2777	2782	2783	2785	2786	2788	2789	2803	2807	2808	2809
2810	2811	2816	2817	2819	2820	2834	2837	2838	2839	2840	2843	2844	2846	2847
2864	2865	2866	2867	2868	2874	2876	2877	2881	2882	2883	2884	2889	2890	2892
2893	2907	2910	2911	2912	2913	2917	2919	2920	2932	2938	2939	2940	2941	2942
2944	2946	2947	2965	2966	2967	2968	2969	2977	2978	2980	2981	2986	2987	2988
2989	2990	2994	2997	2998	3000	3001	3016	3020	3021	3022	3023	3024	3027	3028
3030	3031	3048	3052	3053	3055	3056	3075	3076	3077	3078	3079	3086	3088	3089
3093	3094	3095	3096	3100	3102	3104	3105	3120	3123	3124	3125	3126	3128	3130
3131	3148	3151	3153	3154	3173	3174	3175	3180	3183	3181	3182	3184	3187	3192
3193	3194	3195	3198	3199	3200	3201	3206	3208	3209	3228	3245	3249	3250	3251
3252	3255	3256	3257	3259	3262	3267	3268	3269	3270	3274	3276	3277	3296	3313
3317	3318	3319	3320	3332	3333	3335	3338	3344	3347	3348	3353	3360	3367	3369
3371	3377	3379	3380	3432	3557	3571	3579	3627	3635	3642	3648	3658	3673	3687
3695	3701	3708	3727	3735	3812	3879	3882	3898	3898	3903	3904	3905	3913	3918
3921	3924	3939	3968	3971	3974	3978	3984	3986	3997	4000	4001	4002	4004	4006
4010	4015	4017	4018	4021	4022	4025	4026	4028	4056	4092	4096	4124	4125	4132
4134	4137	4139	4155	4161	4165	4167	4200	4203	4232	4282	4283	4286	4313	4328
4339	4348	4349	4355	4361	4362	4372	4379	4382	4388	4391	4410	4411	4412	4413

	4414	4415	4416	4417	4418	4419	4420	4421	4422	105	106	107	108	127	128
.EQUIV	45	46	54	99	100	101	102	103	104						
.EVEN	129	130	131	132	133	134	135	136							
.GLOBL	236	498	3353	3360	3367	3380	3432	3567	3571	3579	3627	3635	3642	3648	3658
.IF	3673	3687	3695	3701	3708	3727	3967	4331	4378	4513					
	1	21	22	23	24	33	43	109	137	159	162	164	170	172	179
	1923	196	198	227	228	232	233	235	258	266	269	272	275	276	277
	278	279	280	282	407	410	415	417	419	421	423	425	426	443	483
	484	485	488	497	506	508	510	511	512	518	520	522	523	524	525
	527	529	530	531	538	544	546	548	549	550	551	552	553	554	561
	571	573	575	577	579	581	582	583	589	610	612	614	616	617	618
	619	620	626	647	649	651	653	655	657	658	659	665	686	688	690
	692	693	694	695	696	702	723	725	727	729	730	731	732	733	737
	739	741	742	743	745	747	749	750	751	768	771	772	773	774	775
	789	791	792	793	794	795	808	811	813	815	816	817	830	833	834
	835	836	837	842	852	854	856	857	859	861	862	863	867	868	869
	870	871	873	874	875	876	877	893	908	912	914	916	917	918	931
	934	935	936	937	938	951	957	969	971	978	979	981	983	984	984
	985	989	991	993	994	995	996	997	998	999	1000	1008	1015	1017	1019
	1020	1021	1022	1023	1024	1025	1026	1029	1033	1038	1040	1045	1047	1048	1050
	1052	1053	1054	1058	1060	1062	1063	1064	1065	1066	1067	1068	1069	1077	1084
	1086	1088	1089	1090	1091	1092	1093	1094	1095	1098	1102	1107	1109	1114	1116
	1117	1119	1121	1122	1123	1127	1129	1131	1132	1133	1134	1135	1136	1137	1138
	1146	1153	1155	1157	1158	1159	1160	1161	1162	1163	1164	1167	1171	1176	1178
	1183	1185	1186	1188	1190	1191	1192	1196	1198	1200	1201	1202	1203	1204	1205
	1206	1207	1215	1222	1224	1226	1227	1228	1229	1230	1231	1232	1233	1236	1240
	1245	1247	1252	1254	1255	1257	1259	1260	1261	1265	1267	1269	1270	1271	1272
	1273	1274	1275	1276	1284	1291	1293	1295	1296	1297	1298	1299	1300	1301	1302
	1305	1309	1314	1316	1321	1323	1324	1326	1328	1329	1330	1334	1336	1338	1339
	1340	1341	1342	1343	1344	1345	1353	1360	1362	1364	1365	1366	1367	1368	1369
	1370	1371	1374	1378	1383	1385	1390	1392	1393	1395	1397	1398	1399	1403	1405
	1407	1408	1409	1410	1411	1412	1413	1414	1422	1429	1431	1433	1434	1435	1436
	1437	1438	1439	1440	1443	1447	1452	1454	1459	1461	1462	1464	1466	1467	1468
	1472	1474	1476	1477	1478	1479	1480	1481	1482	1483	1491	1498	1500	1502	1503
	1504	1505	1506	1507	1508	1509	1512	1516	1521	1523	1528	1530	1531	1532	1533
	1534	1535	1539	1540	1541	1542	1543	1544	1546	1548	1549	1550	1558	1565	1566
	1567	1568	1569	1570	1572	1574	1575	1576	1579	1583	1588	1590	1595	1597	1598
	1599	1600	1601	1602	1606	1607	1608	1609	1610	1611	1613	1615	1616	1617	1625
	1632	1633	1634	1635	1636	1637	1639	1641	1642	1643	1646	1650	1655	1657	1662
	1664	1665	1666	1667	1668	1669	1673	1674	1675	1676	1677	1678	1680	1682	1683
	1684	1692	1699	1700	1701	1702	1703	1704	1706	1708	1709	1710	1713	1717	1722
	1724	1729	1731	1732	1733	1734	1735	1736	1740	1741	1742	1743	1744	1745	1747
	1749	1750	1751	1759	1766	1767	1768	1769	1770	1771	1773	1775	1776	1777	1780
	1784	1789	1791	1796	1798	1799	1800	1801	1802	1803	1807	1808	1809	1810	1811
	1812	1814	1816	1817	1818	1826	1833	1834	1835	1836	1837	1838	1840	1842	1843
	1844	1847	1851	1856	1858	1863	1865	1866	1867	1868	1869	1870	1874	1875	1876
	1877	1878	1879	1881	1883	1884	1885	1893	1900	1901	1902	1903	1904	1905	1907
	1909	1910	1911	1914	1918	1923	1925	1930	1932	1933	1934	1935	1936	1937	1941
	1942	1943	1944	1945	1946	1948	1950	1951	1952	1960	1967	1968	1969	1970	1971
	1972	1974	1976	1977	1978	1981	1985	1990	1992	1997	1999	2000	2001	2002	2003
	2004	2008	2009	2010	2011	2012	2013	2015	2017	2018	2019	2020	2034	2035	2036
	2037	2038	2039	2041	2043	2044	2045	2048	2052	2057	2059	2061	2064	2066	2068
	2069	2070	2071	2072	2073	2074	2075	2090	2093	2094	2095	2096	2097	2099	2112



2114	2116	2119	2120	2121	2122	2123	2124	2125	2126	2127	2128	2143	2145	2147
2149	2150	2151	2153	2155	2167	2169	2172	2174	2176	2177	2178	2179	2180	2181
2182	2183	2187	2188	2198	2200	2205	2209	2220	2226	22210	22211	22212	22213	22214
2215	2216	2220	2221	2223	2223	2223	2225	2227	2229	22243	22244	22245	22246	22247
2248	2249	2253	2254	2254	2266	2266	2271	2273	2275	2276	2277	2278	2279	2280
2281	2282	2286	2287	2297	2299	2301	2304	2306	2308	2309	2310	2311	2312	2313
2314	2315	2319	2320	2320	2323	2324	2327	2329	2330	2332	2333	2334	2335	2336
2347	2348	2352	2353	2353	2353	2357	2370	2372	2374	2375	2376	2377	2378	2379
2380	2381	2385	2386	2386	2386	2390	2403	2405	2407	2408	2409	2410	2411	2412
2413	2414	2418	2419	2419	2421	2422	2430	2438	2440	2441	2442	2443	2444	2445
2446	2447	2451	2452	2452	2457	2457	2470	2471	2472	2473	2474	2475	2477	2479
2480	2481	2485	2486	2486	2497	2499	2500	2503	2504	2505	2506	2507	2509	2511
2512	2513	2517	2518	2518	2529	2529	2534	2537	2538	2537	2538	2539	2541	2543
2544	2545	2549	2550	2550	2551	2551	2556	2557	2558	2559	2570	2571	2573	2575
2576	2577	2581	2582	2582	2583	2583	2586	2589	2590	2591	2600	2603	2605	2607
2608	2609	2613	2614	2614	2617	2617	2620	2623	2624	2625	2634	2635	2637	2639
2640	2641	2645	2646	2646	2647	2647	2652	2653	2654	2655	2666	2667	2669	2671
2672	2673	2677	2678	2678	2679	2679	2681	2683	2684	2685	2698	2699	2701	2703
2704	2705	2709	2710	2710	2719	2723	2726	2727	2728	2729	2730	2731	2733	2735
2736	2737	2741	2742	2742	2753	2757	2758	2763	2765	2767	2768	2769	2772	2774
2776	2777	2788	2789	2789	2793	2795	2798	2800	2801	2805	2807	2809	2810	2811
2814	2816	2818	2819	2819	2827	2833	2837	2838	2839	2840	2841	2843	2845	2846
2847	2863	2865	2867	2867	2873	2874	2875	2876	2877	2880	2881	2882	2883	2884
2888	2889	2890	2891	2891	2893	2894	2895	2896	2897	2898	2899	2916	2917	2918
2919	2920	2933	2936	2936	2940	2941	2942	2943	2944	2945	2946	2947	2964	2966
2968	2969	2975	2977	2977	2980	2981	2984	2986	2988	2989	2990	2993	2995	2997
3050	3000	3001	3015	3015	3020	3020	3024	3024	3028	3027	3029	3030	3031	3047
3093	3052	3054	3055	3055	3074	3074	3078	3079	3085	3086	3087	3088	3089	3092
3126	3127	3128	3129	3129	3131	3131	3147	3104	3105	3119	3122	3123	3124	3125
3178	3180	3182	3184	3184	3191	3191	3192	3151	3152	3153	3154	3172	3174	3176
3205	3206	3207	3208	3208	3227	3227	3244	3250	3251	3252	3253	3257	3259	3260
3266	3267	3268	3269	3269	3273	3273	3274	3276	3277	3277	3277	3277	3277	3277
3331	3332	3333	3334	3334	3337	3337	3346	3348	3352	3353	3354	3369	3371	3377
3379	3380	3431	3566	3570	3578	3578	3626	3641	3647	3657	3672	3686	3694	3700
3707	3726	3734	3811	3878	3881	3881	3891	3901	3903	3904	3906	3913	3917	3918
3921	3923	3938	3954	3970	3973	3973	3977	3984	3996	3998	3999	4000	4002	4003
4004	4006	4008	4016	4018	4020	4022	4021	4024	4026	4027	4028	4056	4095	4096
4124	4132	4133	4137	4138	4154	4154	4155	4164	4167	4179	4202	4223	4281	4283
4286	4313	4328	4338	4348	4349	4354	4361	4362	4370	4372	4376	4381	4387	4391
4402	4411	4412	4413	4414	4415	4416	4418	4419	4420	4421	4422			
21	23	24	43	160	164	166	171	173	180	193	196	199	227	233
236	415	483	484	506	507	508	509	511	518	521	525	528	539	545
546	547	548	549	551	552	552	573	574	575	577	580	590	611	612
613	614	617	648	648	649	649	651	653	656	666	687	688	689	690
693	703	724	725	726	727	730	737	740	745	748	769	772	790	792
809	811	814	831	834	843	853	854	855	856	857	860	868	874	894
909	912	915	932	935	952	952	970	971	977	978	979	982	989	992
997	1008	1015	1018	1023	1030	1033	1039	1040	1046	1047	1048	1051	1058	1061
1066	1077	1084	1087	1092	1099	1102	1108	1109	1115	1116	1117	1120	1127	1130
1135	1146	1153	1156	1161	1168	1171	1177	1178	1184	1185	1186	1189	1196	1199
1204	1215	1222	1225	1230	1237	1240	1246	1247	1253	1254	1255	1258	1265	1268
1273	1284	1291	1294	1299	1306	1309	1315	1316	1322	1323	1324	1327	1334	1337
1342	1353	1360	1363	1368	1375	1378	1384	1385	1391	1392	1393	1396	1403	1406

. IFF

	1411	1422	1429	1432	1437	1444	1447	1453	1454	1460	1461	1462	1465	1472	1475
	1480	1491	1498	1501	1506	1513	1516	1522	1523	1529	1530	1532	1540	1544	1547
	1559	1566	1570	1573	1580	1584	1589	1590	1596	1597	1599	1607	1611	1614	1626
	1633	1637	1640	1647	1651	1656	1657	1663	1664	1666	1674	1678	1681	1693	1700
	1704	1707	1714	1718	1723	1724	1730	1731	1733	1741	1745	1748	1760	1767	1771
	1774	1781	1785	1790	1791	1797	1798	1800	1808	1812	1815	1827	1834	1838	1841
	1848	1852	1857	1858	1864	1865	1867	1875	1879	1882	1894	1901	1905	1908	1915
	1919	1924	1925	1931	1932	1934	1942	1946	1949	1961	1968	1972	1975	1982	1986
	1991	1992	1998	1999	2001	2009	2013	2016	2028	2035	2039	2042	2049	2053	2058
	2059	2060	2061	2064	2067	2072	2091	2094	2100	2113	2114	2115	2116	2120	2125
	2144	2145	2148	2154	2166	2167	2168	2169	2172	2175	2180	2188	2193	2199	2200
	2201	2202	2205	2208	2213	2221	2226	2232	2233	2234	2235	2238	2241	2246	2254
	2259	2265	2266	2267	2268	2271	2274	2279	2287	2292	2298	2299	2300	2301	2304
	2307	2312	2320	2325	2331	2332	2333	2334	2337	2340	2345	2353	2358	2364	2365
	2366	2367	2370	2373	2378	2386	2391	2397	2398	2399	2400	2403	2406	2411	2419
	2424	2430	2431	2432	2433	2436	2439	2444	2452	2457	2464	2465	2466	2467	2471
	2474	2478	2486	2497	2498	2501	2507	2510	2517	2518	2528	2528	2529	2530	2531
	2533	2539	2542	2550	2560	2561	2562	2563	2567	2571	2574	2582	2592	2593	2594
	2595	2599	2603	2606	2614	2624	2625	2626	2627	2631	2635	2638	2646	2656	2657
	2658	2659	2663	2667	2670	2678	2688	2689	2690	2691	2695	2699	2702	2710	2720
	2721	2722	2723	2727	2731	2734	2742	2754	2755	2756	2757	2763	2766	2772	2775
	2783	2786	2803	2805	2808	2814	2817	2834	2837	2841	2844	2864	2865	2866	2867
	2874	2881	2888	2890	2907	2910	2917	2936	2937	2939	2944	2965	2966	2967	2968
	2975	2978	2984	2987	2994	2995	2998	3016	3018	3021	3025	3028	3048	3050	3053
	3075	3076	3077	3078	3086	3093	3099	3102	3120	3123	3128	3148	3151	3173	3174
	3175	3176	3178	3181	3192	3198	3206	3228	3245	3249	3250	3251	3252	3253	3256
	3267	3274	3296	3313	3317	3318	3319	3320	3332	3334	3337	3344	3347	3379	3735
	3812	3879	3881	3894	3917	3918	3924	3939	3968	3971	3996	4000	4001	4004	4021
	4025	4028	4096	4098	4103	4124	4125	4134	4138	4155	4165	4203	4282	4339	4355
	4372	4382	4388												
. IFT	498	3353	3360	3367	3432	3567	3571	3579	3627	3635	3642	3648	3658	3673	3687
. IFTF	3695	3701	3708	3727	3904	4006	4098	4103	4183	4199	4200				
. IIF	498	3353	3360	3367	3432	3567	3571	3579	3627	3635	3642	3648	3658	3673	3687
	3695	3701	3708	3727	3903	4004	4043	4096	4099	4179	4183	4199			
	425	7	12	13	18	19	20	21	23	24	30	232	236	416	419
	425	426	427	484	3333	3338	3339	3355	3362	3379	3380	3575	3581	3644	3654
	3697	3703	3882	3883	3884	3885	3886	3890	3916	3918	3921	3936	3961	3974	3975
	3976	3977	3978	3982	4005	4018	4021	4022	4025	4046	4147	4155	4161	4279	4410
. IRP	4411	4412	4413	4414	4416	4418	4419	4420	4421						
	407	506	544	571	610	647	686	723	852	969	1038	1107	1176	1245	1314
	1383	1452	1521	1588	1655	1722	1789	1856	1923	1990	2057	2112	2165	2198	2231
	2264	2297	2330	2363	2396	2429	2463	2495	2527	2559	2591	2623	2655	2687	2719
	2753	2863	2964	3074	3172	3248	3316	3337	3822	3862	3918	3983	4174	4195	4287
	4288	4309	4325	4326	4342	4348	4361	4362							
. LIST	1	23	30	151	227	233	236	407	427	484	485	498	506	510	539
	542	544	548	565	568	571	575	590	593	606	609	610	614	627	630
	643	646	647	651	666	669	682	685	686	690	703	706	719	722	723
	727	759	762	765	768	781	784	786	789	800	803	805	808	822	825
	827	830	843	846	848	851	852	856	884	887	890	893	900	903	905
	908	923	926	928	931	943	946	948	951	958	961	963	966	969	978
	1005	1008	1010	1013	1030	1033	1034	1037	1038	1047	1074	1077	1079	1082	1099
	1102	1103	1106	1107	1116	1143	1146	1148	1151	1168	1171	1172	1175	1176	1185
	1212	1215	1217	1220	1237	1240	1241	1244	1245	1254	1281	1284	1286	1289	1306
	1309	1310	1313	1314	1323	1350	1353	1355	1358	1375	1378	1379	1382	1383	1392

1419	1422	1424	1427	1444	1447	1448	1451	1452	1461	1488	1491	1493	1496	1513
1516	1517	1520	1521	1530	1555	1558	1560	1563	1580	1583	1584	1587	1588	1597
1622	1625	1627	1630	1647	1650	1651	1654	1655	1664	1689	1692	1694	1697	1714
1717	1718	1721	1722	1731	1756	1759	1761	1764	1781	1784	1785	1788	1789	1798
1823	1826	1828	1831	1848	1851	1852	1855	1856	1865	1890	1893	1895	1898	1915
1918	1919	1922	1923	1932	1957	1960	1962	1965	1982	1985	1986	1989	1990	1999
2024	2027	2029	2032	2049	2052	2053	2056	2057	2061	2080	2083	2087	2090	2103
2106	2108	2111	2112	2116	2133	2136	2140	2143	2158	2161	2165	2169	2189	2192
2194	2197	2198	2202	2222	2225	2227	2230	2231	2235	2255	2258	2260	2263	2264
2268	2288	2291	2293	2296	2297	2301	2321	2324	2326	2329	2330	2334	2354	2357
2359	2362	2363	2367	2387	2390	2392	2395	2396	2400	2420	2423	2425	2428	2429
2433	2453	2456	2458	2461	2463	2467	2486	2489	2491	2494	2495	2499	2518	2521
2523	2526	2527	2531	2550	2553	2555	2558	2559	2563	2582	2585	2587	2590	2591
2595	2614	2617	2619	2622	2623	2627	2646	2649	2651	2654	2655	2659	2678	2681
2683	2686	2687	2691	2710	2713	2715	2718	2719	2723	2742	2745	2747	2750	2753
2757	2793	2796	2798	2801	2824	2827	2829	2832	2852	2855	2858	2861	2863	2867
2897	2900	2902	2905	2924	2927	2929	2933	2935	2955	2958	2961	2964	2968	3005
3008	3011	3014	3037	3040	3043	3046	306	3065	3068	3071	3074	3078	3109	3112
3115	3118	3137	3140	3143	3146	3160	3163	3166	3169	3172	3176	3215	3218	3223
3226	3235	3238	3241	3244	3248	3252	3283	3286	3291	3294	3303	3306	3309	3312
3316	3320	3338	3353	3360	3367	3371	3432	3445	3448	3452	3455	3567	3571	3579
3627	3635	3642	3648	3658	3673	3687	3695	3701	3708	3727	3918	3977	4124	4402
4410	4411	4412	4413	4414	4415	4416	4417	4418	4419	4420	4421	4422		
.MACRO	24	156	190	361	443	569	967	969	1038	1107	1176	1245	1314	1383
	1452	1521	1588	1655	1722	1789	1856	1923	2164	2752	2963	3329	3876	4402
.MCALL	1	151	233	427	485									
.MEXIT	281													
.NLIST	1	23	30	151	227	233	236	407	427	484	498	506	510	539
	542	544	548	565	568	571	575	590	593	606	610	614	627	630
	643	646	647	651	666	669	682	685	686	690	706	719	722	723
	727	759	762	765	768	781	784	786	789	800	805	808	822	825
	827	830	843	846	848	851	852	856	884	887	893	900	903	905
	908	923	926	928	931	943	946	948	951	958	963	966	969	978
	1005	1008	1010	1013	1030	1033	1034	1037	1038	1047	1074	1079	1082	1099
	1102	1103	1106	1107	1116	1143	1146	1148	1151	1168	1171	1175	1176	1185
	1212	1215	1217	1220	1237	1240	1241	1244	1245	1254	1281	1284	1289	1306
	1309	1310	1313	1314	1323	1350	1353	1355	1358	1375	1378	1379	1382	1392
	1419	1422	1424	1427	1444	1447	1448	1451	1452	1461	1488	1491	1493	1513
	1516	1517	1520	1521	1530	1555	1558	1560	1563	1580	1583	1584	1587	1597
	1622	1625	1627	1630	1647	1650	1651	1654	1655	1664	1689	1692	1694	1714
	1717	1718	1721	1722	1731	1756	1759	1761	1764	1781	1784	1785	1788	1798
	1823	1826	1828	1831	1848	1851	1852	1855	1856	1865	1890	1893	1895	1915
	1918	1919	1922	1923	1932	1957	1960	1962	1965	1982	1985	1986	1989	1999
	2024	2027	2029	2032	2049	2052	2053	2056	2057	2061	2080	2083	2087	2103
	2106	2108	2111	2112	2116	2133	2136	2140	2143	2158	2161	2165	2189	2192
	2194	2197	2198	2202	2222	2225	2227	2230	2231	2235	2255	2258	2260	2264
	2268	2288	2291	2293	2296	2297	2301	2321	2324	2326	2330	2334	2354	2357
	2359	2362	2363	2367	2387	2390	2392	2395	2396	2400	2420	2423	2425	2429
	2433	2453	2456	2458	2461	2463	2467	2486	2489	2491	2494	2495	2499	2521
	2523	2526	2527	2531	2550	2553	2555	2558	2559	2563	2582	2585	2587	2591
	2595	2614	2617	2619	2622	2623	2627	2646	2649	2651	2654	2655	2659	2681
	2683	2686	2687	2691	2710	2713	2715	2718	2719	2723	2742	2745	2747	2753
	2757	2793	2796	2798	2801	2824	2827	2829	2832	2852	2855	2861	2863	2867
	2897	2900	2902	2905	2924	2927	2929	2933	2935	2955	2961	2964	2968	3005

	3008	3011	3014	3037	3040	3043	3046	3062	3065	3068	3071	3074	3078	3109	3112
	3115	3118	3137	3140	3143	3146	3160	3163	3166	3169	3172	3176	3215	3218	3223
	3226	3235	3238	3241	3244	3248	3252	3283	3286	3291	3294	3303	3306	3309	3312
	3316	3320	3338	3353	3360	3367	3371	3432	3445	3448	3452	3455	3567	3571	3579
	3627	3635	3642	3648	3658	3673	3687	3695	3701	3708	3727	3918	3977	4124	4402
	4410	4411	4412	4413	4414	4415	4416	4417	4418	4419	4420	4421	4422		
.PAGE	190														
.PSECT	1														
.REPT	30														
.SBTTL	14	24	41	157	168	190	233	282	409	480	485	506	544	571	610
	647	686	723	852	969	1038	1107	1176	1245	1314	1383	1452	1521	1588	1655
	1722	1789	1856	1923	1990	2057	2112	2165	2198	2231	2264	2297	2330	2363	2396
	2429	2463	2495	2527	2559	2591	2623	2655	2687	2719	2753	2863	2964	3074	3172
	3248	3316	3329	3732	3809	3876	3921	3968	4022	4162	4200	4279	4336	4379	4402
.TITLE	2														
.WORD	30	31	32	165	184	185	186	187	188	189	198	201	202	203	204
	207	208	209	210	211	212	213	216	217	218	238	239	240	241	242
	243	244	245	249	250	251	264	268	271	274	275	276	277	278	279
	373	374	375	376	377	378	379	380	381	382	383	385	389	390	391
	392	393	394	395	396	397	398	399	401	402	403	404	405	520	523
	527	530	553	579	582	619	655	658	695	732	739	742	747	750	774
	794	813	816	836	859	862	870	876	914	917	937	981	984	991	994
	999	1017	1020	1025	1050	1053	1060	1063	1068	1086	1089	1094	1119	1122	1129
	1132	1137	1155	1158	1163	1188	1191	1198	1201	1206	1224	1227	1232	1257	1260
	1267	1270	1275	1293	1296	1301	1326	1329	1336	1339	1344	1362	1365	1370	1395
	1398	1405	1408	1413	1431	1434	1439	1464	1467	1474	1477	1482	1500	1503	1508
	1534	1542	1546	1549	1568	1572	1575	1601	1609	1613	1616	1635	1639	1642	1668
	1676	1680	1683	1702	1706	1709	1735	1743	1747	1750	1769	1773	1776	1802	1810
	1814	1817	1836	1840	1843	1869	1877	1881	1884	1903	1907	1910	1936	1944	1948
	1951	1970	1974	1977	2003	2011	2015	2018	2037	2041	2044	2066	2069	2074	2096
	2122	2127	2147	2150	2174	2177	2182	2207	2210	2215	2240	2243	2248	2273	2276
	2281	2306	2309	2314	2339	2342	2347	2372	2375	2380	2405	2408	2413	2438	2441
	2446	2473	2477	2480	2505	2509	2512	2537	2541	2544	2559	2573	2576	2601	2605
	2608	2633	2637	2640	2665	2669	2672	2697	2701	2704	2729	2733	2736	2765	2768
	2774	2777	2785	2788	2807	2810	2816	2819	2839	2843	2846	2876	2883	2892	2912
	2919	2938	2941	2946	2977	2980	2986	2989	2997	3000	3020	3023	3027	3030	3052
	3055	3088	3095	3104	3125	3130	3153	3180	3184	3185	3186	3194	3200	3208	3255
	3259	3260	3261	3269	3276	3343	3346	3378	3514	3808	3947	3952	4199	4229	4276
	4311	4371	4409	4515	4517	4519	4524	4525	4526						

000000

ERRORS DETECTED: 0

H10

MD-11-DRLPN-A MACY11 27(654) 15-DEC-77 08:43 PAGE 112  
DRLPN.P11

SEQ 0124

\*DRLPN,DRLPN/CRF/SOL=DRLPN  
RUN-TIME: 36 26 4 SECONDS  
CORE USED: 28K  
EOF1DRLPNASEQ

00010000

780223

PDP10 411