M8254

(IPBM) FIELD DIAGNOSTIC

MD-11-DRLPN-A

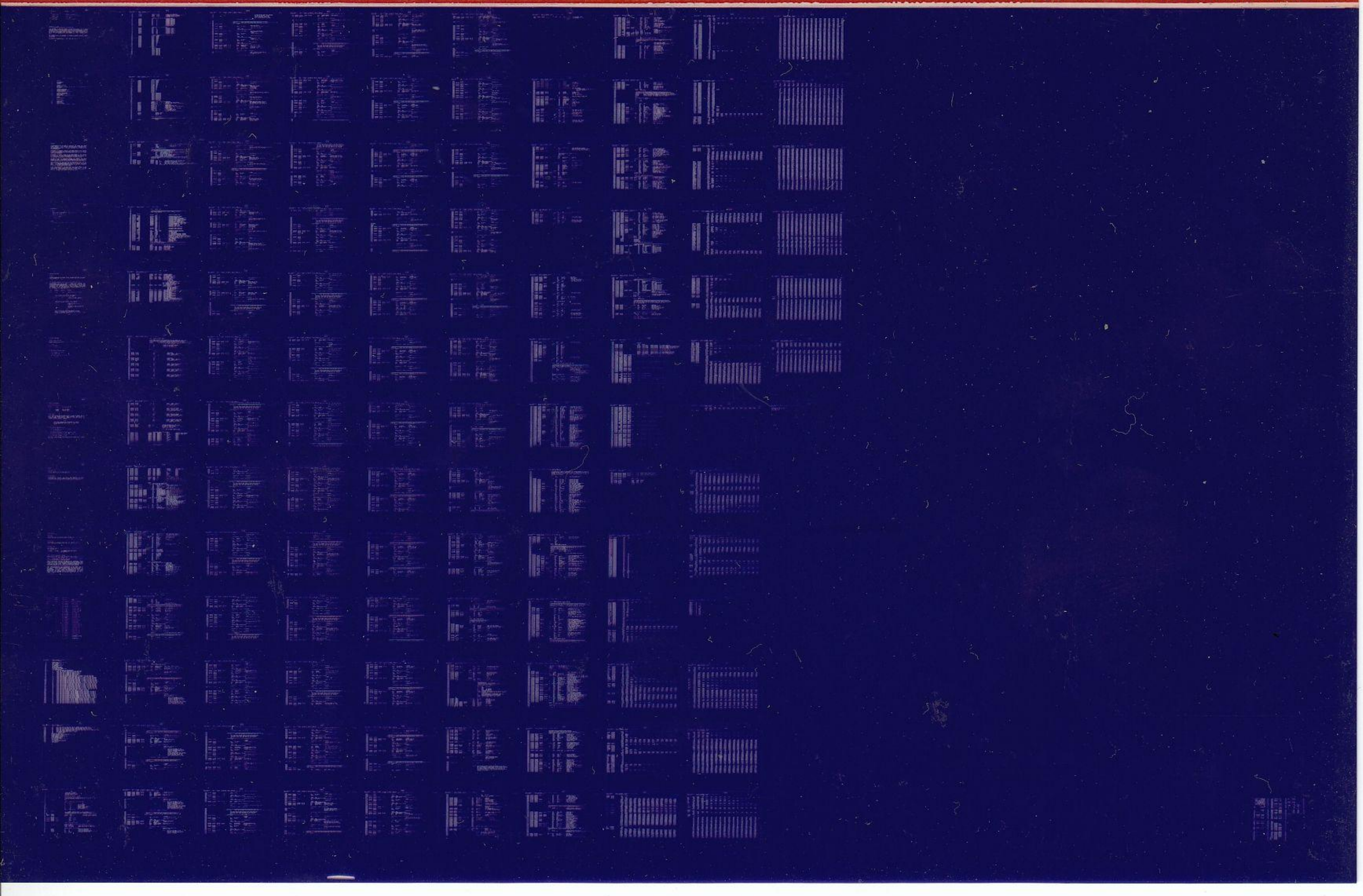
EP-DRLPN-A-DL

FICHE 1 OF 1

MAR 1978

digital

MADE IN USA



HDR1DRLPNASEQ 00010000 780223 PDP10 411 IDENTIFICATION SEQ 0001

PRODUCT CODE: MAINDEC-11-DRLPN-A-D

PRODUCT NAME: M8254 (IPBM) FIELD DIAGNOSTIC

DATE CREATED: JAN. 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

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 ORIGINALY 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 HAVE 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.

SEQ 0004

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 VARIENCE 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.

TAG	ADDRESS	CONTENTS	COMMENTS
$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	;VARIENCE IN ADDR. BETWEEN ; KMC 1 AND THE M8200-YC.

NOTE

THIS LOCATION MUST REFLECT THE VARIENCE IN ADDRESS THE M8200-YC HAS OVER THE KMC-11.

MCNT:	001366	1	;NUMBER OF M8254 MODULES TO ; CHECK OUT
VAR2:	001370	20	;VARIENCE 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

SEQ 0006

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 M8254 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 "WATCHS" THE KMC-11 AND M8200-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 M8200-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. ALTOUGH "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:

OPTION	GROUP	DIAG. #	DIAG. TITLE
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
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

```
                                .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.
                                ;*
000001                          $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

000000                                  .=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
000174                                  .=174
000174  000000         DISPREG: .WORD  0               ;;SOFTWARE DISPLAY REGISTER
000176  000000         SWREG:   .WORD  0               ;;SOFTWARE SWITCH REGISTER

000200                                  .=200
000200  000137  001464                JMP     START
000210                                  .=210
000210  000137  012632                JMP     MICRAM
000220                                  .=220
000220  000137  013070                JMP     ODT

                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100          STACK=  1100
                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                ;*MISCELLANEOUS DEFINITIONS
000011          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012          LF=     12              ;;CODE FOR LINE FEED
000015          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776          PS=     177776          ;;PROCESSOR STATUS WORD
                .EQUIV  PS,PSW
```

```
                        
                        STKLMT= 177774          ;;STACK LIMIT REGISTER
                        PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
                        DSWR=   177570          ;;HARDWARE SWITCH REGISTER
                        DDISP=  177570          ;;HARDWARE DISPLAY REGISTER
                        
                        ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                  R0=     %0              ;;GENERAL REGISTER
000001                  R1=     %1              ;;GENERAL REGISTER
000002                  R2=     %2              ;;GENERAL REGISTER
000003                  R3=     %3              ;;GENERAL REGISTER
000004                  R4=     %4              ;;GENERAL REGISTER
000005                  R5=     %5              ;;GENERAL REGISTER
000006                  R6=     %6              ;;GENERAL REGISTER
000007                  R7=     %7              ;;GENERAL REGISTER
000006                  SP=     %6              ;;STACK POINTER
000007                  PC=     %7              ;;PROGRAM COUNTER
                        
                        ;*PRIORITY LEVEL DEFINITIONS
000000                  PR0=    0               ;;PRIORITY LEVEL 0
000040                  PR1=    40              ;;PRIORITY LEVEL 1
000100                  PR2=    100             ;;PRIORITY LEVEL 2
000140                  PR3=    140             ;;PRIORITY LEVEL 3
000200                  PR4=    200             ;;PRIORITY LEVEL 4
000240                  PR5=    240             ;;PRIORITY LEVEL 5
000300                  PR6=    300             ;;PRIORITY LEVEL 6
000340                  PR7=    340             ;;PRIORITY LEVEL 7
                        
                        ;*"SWITCH REGISTER" SWITCH DEFINITIONS
100000                  SW15=   100000
040000                  SW14=   40000
020000                  SW13=   20000
010000                  SW12=   10000
004000                  SW11=   4000
002000                  SW10=   2000
001000                  SW09=   1000
000400                  SW08=   400
000200                  SW07=   200
000100                  SW06=   100
000040                  SW05=   40
000020                  SW04=   20
000010                  SW03=   10
000004                  SW02=   4
000002                  SW01=   2
000001                  SW00=   1
                        .EQUIV  SW09,SW9
                        .EQUIV  SW08,SW8
                        .EQUIV  SW07,SW7
                        .EQUIV  SW06,SW6
                        .EQUIV  SW05,SW5
                        .EQUIV  SW04,SW4
                        .EQUIV  SW03,SW3
                        .EQUIV  SW02,SW2
                        .EQUIV  SW01,SW1
                        .EQUIV  SW00,SW0
```

```
                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                  BIT15=  100000
040000                  BIT14=  40000
020000                  BIT13=  20000
010000                  BIT12=  10000
004000                  BIT11=  4000
002000                  BIT10=  2000
001000                  BIT09=  1000
000400                  BIT08=  400
000200                  BIT07=  200
000100                  BIT06=  100
000040                  BIT05=  40
000020                  BIT04=  20
000010                  BIT03=  10
000004                  BIT02=  4
000002                  BIT01=  2
000001                  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
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR

170460                          ABASE=  170460
000300                          AVECT1= 300
000300                          APRIOR= 300
000001                          $TN=1

                        .SBTTL  ACT11 HOOKS

                        ;;**********************************************************************
                        ;HOOKS REQUIRED BY ACT11
000224                          $SVPC=.                 ;SAVE PC
000046                          .=46
```

```
                                        $ENDAD                  ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOp
                                        .=52
                                        .WORD   0               ;;2)SET LOC.52 TO ZERO
                                        .=$SVPC                 ;; RESTORE PC
                                        .=1000
                                .SBTTL  APT PARAMETER BLOCK

                                ;;**************************************************************
                                ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                                ;;**************************************************************
                                        .$X=.   ;;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.
                                        $APTHDR ;;POINT TO APT HEADER BLOCK
                                        .=.$X   ;;RESET LOCATION COUNTER
                                ;;**************************************************************
                                ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC
                                ;INTERFACE SPEC.

                                $APTHD:
                                $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
                                $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
                                $TSTM:  .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   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000          $TSTNM: .WORD   0               ;;CONTAINS THE TEST NUMBER
001102  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001103  000             $ICNT:  .BYTE   0               ;;CONTAINS SUBTEST ITERATION COUNT
001104  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001106  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001110  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001112  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001113  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001114  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001116  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001120  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001122  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001124  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001126  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001130  000000                  .WORD   0
001132  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001133  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001134  000000                  .WORD   0
001136  177570          $SWR:   .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001140  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001142  177560          $TKS:   177560                  ;;TTY KBD STATUS
001144  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001146  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001150  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001152  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001153  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001154  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001155  000             $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001156  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001160  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001164  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001165  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001166  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001170                  $MAIL:                          ;;APT MAILBOX
001170  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001172  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001174  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001176  000000          $PASS:  .WORD   APASS           ;;PASS COUNT
001200  000000          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001202  000000          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
```

```
001206  000000          $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
001210  000000          $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
001212                  $ETABLE:                ;;APT ENVIRONMENT TABLE
001212     000          $ENv:   .BYTE   AENV    ;;ENVIRONMENT BYTE
001213     000          $ENvM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
001214  000000          $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
001216  000000          $USWR:  .WORD   AUSWR   ;;USER SWITCHES
001220  000000          $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
                        ;*                     BITS 15-11=CPU TYPE
                        ;*                                 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
                        ;*                                 11/70=06,PDQ=07,Q=10
                        ;*                     BIT 10=REAL TIME CLOCK
                        ;*                     BIT  9=FLOATING POINT PROCESSOR
                        ;*                     BIT  8=MEMORY MANAGEMENT
001222     000          $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
001223     000          $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
                        ;*                     MEM.TYPE BYTE   --  (HIGH BYTE)
                        ;*                                 900 NSEC CORE=001
                        ;*                                 300 NSEC BIPOLAR=002
                        ;*                                 500 NSEC MOS=003
001224  000000          $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
                        ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
001226     000          $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
001227     000          $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
001230  000000          $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
001232     000          $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
001233     000          $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
001234  000000          $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
001236     000          $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
001237     000          $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
001240  000000          $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
001242  000300          $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
001244  000000          $VECt2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
001246  170460          $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
001250  000000          $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
001252  000000          $CDW1:  .WORd   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
001254                  $ETEND:
                        .MEXIT
```

```
                                .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 $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($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

001254  016507                          EM1                     ;UBSR ERROR
001256  017102                          DH1                     ;ERRPC   GDDAT   BAD DATA
001260  017250                          DT1                     ;$ERRPC,$GDDAT,$BDDAT
001262  017310                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   2

001264  016523                          EM2                     ;IOSR ERROR
001266  017102                          DH1                     ;ERRPC   GDDAT   BAD DATA
001270  017250                          DT1                     ;$ERRPC,$GDDAT,$BDDAT
001272  017310                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   3

001274  016537                          EM3                     ;SIDE 1 (IO TO UB) FAST PATH DATA ERROR
001276  017102                          DH1                     ;ERRPC   GDDAT   BAD DATA
001300  017250                          DT1                     ;$ERRPC,$GDDAT,$BDDAT
001302  017310                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   4

001304  016607                          EM4                     ;SIDE 2 (UB TO IO) FAST PATH DATA ERROR
001306  017102                          DH1                     ;ERRPC   GDDAT   BAD DATA
001310  017250                          DT1                     ;$ERRPC,$GDDAT,$BDDAT
001312  017310                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   5

001314  016657                          EM5                     ;SIDE 1 (IO TO UB) SILO DATA ERROR
001316  017102                          DH1                     ;ERRPC   GDDAT   BAD DATA
001320  017250                          DT1                     ;$ERRPC,$GDDAT,$BDDAT
001322  017310                          DF0                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   6

001324  016722                          EM6                     ;SIDE 2 (UB TO IO) SILO DATA ERROR
```

```
                                        DH1                     ;ERRPC   GDDAT   BAD DATA
                                        DT1                     ;$ERRPC,$GDDAT,$BDDAT
                                        DFO                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   7

                                        EM7                     ;SIDE 1 INPO DATA ERROR
                                        DH7                     ;ERRPC   INP& #  GDDAT   BDDAT
                                        DT7                     ;$ERRPC,INNU,$GDDAT,$BDDAT
                                        DFO                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   10

                                        EM10                    ;SIDE 2 INPO DATA ERROR
                                        DH7                     ;ERRPC   INPO #  GDDAT   BDDAT
                                        DT7                     ;$ERRPC,INNU,$GDDAT,$BDDAT
                                        DFO                     ;ALL NUMBERS ARE IN OCTAL FORM.

                                        ;ITEM   11

                                        EM11                    ;MICRO-P DETECTED IPBM ERROR
                                        DH11                    ;TEST    ERRPC   KMCADR  CSR
                                        DT11                    ;$TSTNM,$ERRPC,DLAD,$BDDAT
                                        DFO                     ;ALL NUMBERS ARE IN OCTAL FORM.


001364  000010          VAR1:   10              ;VARIENCE BETWEEN KMC 1 AND M8200-YC
001366  000001          MCNT:   1               ;#OF M8254'S TO BE TESTED
001370  000020          VAR2:   20              ;VARIENCE IN ADDRS BETWEEN TEST STATIONS
001372  000001          NCNT:   1               ;CURRENT TEST STATION

                                                ;NOTE: TEST STATION CONSISTS OF A
                                                ;       KMC-11, M8254, AND AN M8200-YC


                                        ;MAINTANCE M8200-YC ADDRESSES
001374  170470          KMMCSR: .WORD   ABASE+10        ;>PATCH         <;M8200-YC BASE ADDRESS
001376  170474          KMMADR: .WORD   ABASE+14        ;>NO            <;M8200-YC CRAM ADDRESS
001400  170476          KMMDBR: .WORD   ABASE+16        ;>PATCHES       <;M8200-YC CRAM DATA
001402  170470          MBSEL0: .WORD   ABASE+10        ;>HERE          <;BSEL0
001404  170471          MBSEL1: .WORD   ABASE+11        ;>NO            <;BSEL1
001406  170472          MBSEL2: .WORD   ABASE+12        ;>PATCHES       <;BSEL2
001410  170473          MBSEL3: .WORD   ABASE+13        ;>HERE          <;BSEL3
001412  170474          MBSEL4: .WORD   ABASE+14        ;>NO            <;BSEL4
001414  170475          MBSEL5: .WORD   ABASE+15        ;>PATCHES       <;BSEL5
001416  170476          MBSEL6: .WORD   ABASE+16        ;>HERE          <;BSEL6
001420  170477          MBSEL7: .WORD   ABASE+17        ;>NO NO         <;BSEL7

001422  177546          KWADR:  .WORD   177546          ;ADDR OF CLOCK.


                                        ;1ST KMC-11 ADDRESS
001424  170460          KMCSR:  .WORD   ABASE           ;>PATCH         <;BASE ADDRESS OF KMC-11
```

```
                                KMADR:  .WORD   ABASE+4         ;>ADDRESS       <;CRAM ADDR. REG BITS0:9
                                KMDBR:  .WORD   ABASE+6         ;>"$BASE"       <;CRAM DATA REG
                                BSEL0:  .WORD   ABASE           ;>IF ADDR.      <;BSEL0
                                BSEL1:  .WORD   ABASE+1         ;>DIFFERENT     <;BSEL1
                                BSEL2:  .WORD   ABASE+2         ;>              <;BSEL2
                                BSEL3:  .WORD   ABASE+3         ;>NO            <;BSEL3
                                BSEL4:  .WORD   ABASE+4         ;>PATCHES       <;BSEL4
                                BSEL5:  .WORD   ABASE+5         ;>HERE          <;BSEL5
                                BSEL6:  .WORD   ABASE+6         ;>NO            <;BSEL6
                                BSEL7:  .WORD   ABASE+7         ;>PATCHES       <;BSEL7

                                CKVECT: .WORD   100             ;VECTOR ADDRESS OF CLOCK.
                                CKVCT2: .WORD   102
                                ERCNT:  .WORD   0               ;TOTAL ERROR COUNT DURING RUN.
                                INNU:   .WORD   0
                                KLAD:   .WORD   0               ;ADDR. OF FAILING KMC FOR CERTAIN TESTS.

001464                          START:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
001464  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
001470  005026                  CLR     (R6)+           ;;CLEAR MEMORY LOCATION
001472  022706  001140          CMP     #SWR,R6 ;;DONE?
001476  001374                  BNE     .-6             ;;LOOP BACK IF NO
001500  012706  001100          MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
001504  012737  014230  000020          MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
001512  012737  000340  000022          MOV     #340,@#IOTVEC+2 ;;LEVEL 7
001520  012737  013726  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
001526  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
001534  012737  016226  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
001542  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
001550  012737  016050  000024          MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
001556  012737  000340  000026          MOV     #340,@#PWRVEC+2 ;;LEVEL 7
001564  005037  001160                  CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
001570  012737  001570  001106          MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
001576  013746  000004          MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
001602  012737  001636  000004  MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
001610  012737  177570  001140  MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
001616  012737  177570  001142  MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
001624  022777  177777  177306  CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
001632  001012                  BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                        ;;AND  THE HARDWARE SWR IS NOT = -1
001634  000403                  BR      65$             ;;BRANCH IF NO TIMEOUT
001636  012716  001644  64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
001642  000002                  RTI
001644  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
001652  012737  000174  001142          MOV     #DISPREG,DISPLAY
001660  012637  000004  66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR

001664  005037  001200                  CLR     $PASS           ;;CLEAR PASS COUNT
```

```
                                        BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
                                        BEQ     67$             ;;YES,USE NON-APT SWITCH
                                        MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
                                67$:
                                RSTART: CLR     ERCNT           ;CLEAR ERROR COUNT.
                                RTNAD:  MOV     $BASE,R0        ;GET BASE ADDRESS.
                                        MOV     #1,NCNT
                                LSTED:  MOV     #KMCSR,R1       ;STORAGE
                                        MOV     R0,(1)+         ;FIX ALL ADDS.
                                        ADD     #4,R0
                                        MOV     R0,(1)+
                                        ADD     #2,R0
                                        MOV     R0,(R1)+
                                        MOV     KMCSR,R0        ;NO FIX BSEL ADDRESS.
                                        MOV     #10,R2          ;DO ALL SEVEN.
                                1$:     MOV     R0,(1)+
                                        ADD     #1,R0
                                        DEC     R2
                                        BNE     1$

                                        MOV     KMCSR,KMMCSR
                                        ADD     VAR1,KMMCSR
                                        MOV     KMMCSR,KMMADR   ;NOW FIX M8200-YC ADDRESS.
                                        MOV     KMMCSR,KMMDBR

                                        ADD     #4,KMMADR
                                        ADD     #6,KMMDBR
                                        MOV     KMMCSR,R0       ;NOW FIX BSEL ADDRESSES.
                                        MOV     #MBSEL0,R1
                                        MOV     #10,R2
                                2$:     MOV     R0,(1)+
                                        ADD     #1,R0
                                        DEC     R2
                                        BNE     2$
                                        RESET

                                .SBTTL  TYPE PROGRAM NAME
                                ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
                                        INC     #-1             ;;FIRST TIME?
                                        BNE     64$             ;;BRANCH IF NO
                                        TYPE    ,65$            ;;TYPE ASCIZ STRING
                                .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
                                        TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
                                        BNE     66$             ;;BRANCH IF YES
                                        CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
                                        BEQ     66$             ;;BRANCH IF YES
                                        CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
                                        BNE     67$             ;;BRANCH IF NO
                                        GTSWR                   ;;GET SOFT-SWR SETTINGS
                                        BR      67$
                                66$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
                                67$:
                                        BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <CRLF>#MD-11-DRLPN-A#<CRLF>
```

```
                                64$:
002160                                  INC     #-1             ;IS THIS THE FIRST TIME THROUGH ?
002160  005227  177777                  BNE     3$              ;NO-BRANCH
002164  001005
002166  005227  177777                  INC     #-1
002172  001002                          BNE     3$
002174  104401  016310                  TYPE    ,MESWCH
002200  004537  012246          3$:     JSR     R5,LOAD         ;LOAD UCODE INTO KMC.

                                ;;*****************************************************************
                                ;*TEST 1         *TEST THE INITIAL CONDITIONS OF THE IPBM UBSR
                                ;;*****************************************************************
002204  000240                  TST1:   NOP
002206  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION
002214  012737  002244  001106          MOV     #1$,$LPADR      ;;SET SCOPE LOOP ADDRESS

002222  112737  000001  001102          MOVB    #1,$TSTNM       ;SET TO TEST #1.
002230  012737  002244  001106          MOV     #1$,$LPADR      ;LOOP AT 1$.
002236  012737  002244  001110          MOV     #1$,$LPERR      ;ERROR LOOP AT 1$.

002244                          1$:
                                                                ;/-MCMD-
002244  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
002250  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                                ;/READ FAST PATH REG.
002252  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
002254  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
002260  001424                          .WORD   KMCSR
        000001                          .MD.=1
                                                                ;/COMMAND=NOP.
002262  000001                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

002264  117737  177154  001126          MOVB    @BSEL5,$BDDAT   ;READ CSR.
002272  105037  001127                  CLRB    $BDDAT+1
002276  012737  000052  001124          MOV     #52,$GDDAT      ;EXPECT BITS 5,3,1 SET, ALL
                                                                ;OTHER BITS CLEAR.
002304  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;CSR OK AT INIT?
002312  001401                          BEQ     TST2            ;;

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

002314  104001                          ERROR   1               ;#1 KMC SIDE STATUS REG ERROR (UBSR)

                                ;;*****************************************************************
                                ;*TEST 2         *TEST THE INITIAL CONDITIONS OF THE IPBM IOSR
                                ;;*****************************************************************
002316  000004                  TST2:   SCOPE
002320  012737  000001  001160          MOV     #1,$TIMES       ;;DO 1 ITERATION

002326  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                          .MD.=1
```

```
                                                        ;/COMMAND=NOP.
002332  000001                                .WORD   .MD.
                                                        ;/RETURN HERE AFTER COMMAND .

002334  117737  177054  001126                MOVB    @MBSEL5,$BDDAT  ;READ IOSR. (IPBM TO SLAVE).
002342  105037  001127                        CLRB    $BDDAT+1
002346  012737  000056  001124                MOV     #56,$GDDAT      ;EXPECT BITS 5,3,2,1 SET, ALL OTHER
                                                                        ;BITS CLEAR.
002354  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;SR SIDE 2 (SLAVE) OK AT INIT?
002362  001401                                BEQ     TST3            ;;

002364  104002                                ERROR   2               ;#2 SIDE (IPBM TO sLAVE) STATUS
                                                                        ;REGISTER ERROR IOSR.

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


                                        ;;**************************************************************
                                        ;*TEST 3        *TEST THAT INP 17 READS ALL ZEROS ON SIDE 1
                                        ;;**************************************************************
002366  000004                          TST3:   SCOPE

                                                        ;/-IN1367-
                                                        ;/-MCMD-
002370  004537  011656                        JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
002374  001424                                .WORD   KMCSR
        000020                                .MD.=20
                                                        ;/READ INP0 17 PUT IN BSEL 3
002376  000020                                .WORD   .MD.
                                                        ;/RETURN HERE AFTER COMMAND.
002400  012737  000017  001460                MOV     #17,INNU        ;/RECORD INP0 NUMBER.
002406  012737  000000  001124                MOV     #0,$GDDAT       ;/EXPECT ALL 0Z.
002414  005037  001126                        CLR     $BDDAT
002420  117737  177014  001126                MOVB    @BSEL3,$BDDAT ;/READ WHAT INP 17 WAS
002426  123737  001124  001126                CMPB    $GDDAT,$BDDAT   ;/WAS IT OK?
002434  001401                                BEQ     TST4            ;;

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

002436  104007                                ERROR   7               ;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 <<**********

                                ;;*****************************************************************
                                ;*TEST 4         *TEST THAT INP 17 READS ALL ZEROS ON SIDE 2
                                ;*****************************************************************
002440  000004          TST4:   SCOPE
                                                                ;/-IN1367-
002442  004537  012100          JSR     R5,KMSIM                ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000020                  .MD.=20
                                                                ;/READ INPO 17 PUT IN BSEL 3
002446  000020                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
002450  012737  000017  001460  MOV     #17,INNU                ;/RECORD INPO NUMBER.
002456  012737  000000  001124  MOV     #0,$GDDAT               ;/EXPECT ALL OZ.
002464  005037  001126          CLR     $BDDAT
002470  117737  176714  001126  MOVB    @MBSEL3,$BDDAT          ;/READ WHAT INP 17 WAS
002476  123737  001124  001126  CMPB    $GDDAT,$BDDAT           ;/WAS IT OK?
002504  001401                  BEQ     TST5                    ;;

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

002506  104010                  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 <<**********

                                ;;*****************************************************************
                                ;*TEST 5         *TEST THAT INP 13 READS ALL ONES ON SIDE 1
                                ;*****************************************************************
002510  000004          TST5:   SCOPE
                                                                ;/-IN1367-
                                                                ;/-MCMD-
002512  004537  011656          JSR     R5,ISSUEC               ;/ISSUE COMMAND TO KMC #1.
002516  001424                  .WORD   KMCSR
        000016                  .MD.=16
                                                                ;/READ INPO 13 PUT IN BSEL3
002520  000016                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
```

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

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

                002560  104007                          ERROR   7               ;ERROR INP 13 SHOULD HAVE
                                                                                ;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 <<**********

                                                ;;**************************************************************
                                                ;*TEST 6         *TEST THAT INP 16 READS ALL ONES ON SIDE 2
                                                ;;**************************************************************
                002562  000004                  TST6:   SCOPE

                                                                                ;/-IN1367-
                002564  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
                        000017                          .MD.=17
                                                                                ;/READ INPO 16 PUT IN BSEL3
                002570  000017                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND.
                002572  012737  000016  001460          MOV     #16,INNU        ;/RECORD INPO NUMBER.
                002600  012737  000377  001124          MOV     #377,$GDDAT     ;/EXPECT ALL OZ.
                002606  005037  001126                  CLR     $BDDAT
                002612  117737  176572  001126          MOVB    @MBSEL3,$BDDAT ;/READ WHAT INP 16 WAS
                002620  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/WAS IT OK?
                002626  001401                          BEQ     TST7            ;;

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

                002630  104010                          ERROR   10              ;ERROR INP 16 SHOULD HAVE
                                                                                ;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.
```

```
                                                                    ;            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 <<**********

                                        ;;************************************************************
                                        ;*TEST 7          *TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
                                        ;;************************************************************
  002632  000004                        TST7:   SCOPE

                                                                    ;READ DATA SIDE 2
  002634  004537  012100                        JSR     R5,KMSIM    ;/DMC SIMULATE KMC INSTR. LIST FOR.
          000006                                .MD.=6
                                                                    ;/READ FAST PATH REG.
  002640  000006                                .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .

  002642  112777  000377  176570                MOVB    #377,@BSEL3 ;SET DATA TO BE WRITTEN
                                                                    ;ISSUE COMMAND TO WRITE IT.
                                                                    ;/-MCMD-
  002650  004537  011656                        JSR     R5,ISSUEC   ;/ISSUE COMMAND TO KMC #1.
  002654  001424                                .WORD   KMCSR
          000007                                .MD.=7
                                                                    ;/WRITE FAST PATH REG.
  002656  000007                                .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
                                                                    ;ALLOW SETTLE TIME
                                                                    ;/-MCMD-
  002660  004537  011656                        JSR     R5,ISSUEC   ;/ISSUE COMMAND TO KMC #1.
  002664  001424                                .WORD   KMCSR
          000001                                .MD.=1
                                                                    ;/COMMAND=NOP.
  002666  000001                                .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .

  002670  117737  176550  001126                MOVB    @BSEL5,$BDDAT   ;READ SIDE #1 UBSR.
  002676  105037  001127                        CLRB    $BDDAT+1
  002702  012737  000072  001124                MOV     #72,$GDDAT      ;NORMAL=52 EXPECT BIT4
                                                                        ;TO BE SET.
  002710  023737  001124  001126                CMP     $GDDAT,$BDDAT   ;UBSR SIDE 1 OK? (BIT4 SET.)
  002716  001402                                BEQ     1$

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

  002720  104001                                ERROR   1               ;SIDE 1 CSR BIT4 SHOULD BE SET
                                                                        ;DATA WRITTEN INTO FAST PATH REG. BUT
                                                                        ;NOT READ ON SIDE 2.

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

```
002722  000471                          BR      TST10           ;;

002724                          1$:
002724  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                          .MD.=1
                                                                ;/COMMAND=NOP.
002730  000001                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
002732  117737  176456  001126          MOVB    @MBSEL5,$BDDAT  ;READ SIDE #2 IOSR
002740  012737  000016  001124          MOV     #16,$GDDAT      ;NORMAL=56, EXPECT BIT5
                                                                ;TO BE CLEAR.
002746  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;IS IT CLEAR?
002754  001402                          BEQ     2$              ;YES-NEXT CHECK.

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

002756  104002                          ERROR   2               ;BIT5 OF SIDE #2 IOSR NOT CLEAR WHEN
                                                                ;SIDE #1 MADE XFER VIA FAST PATH.

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

002760  000452                          BR      TST10           ;;
002762                          2$:                             ;READ FAST PATH SIDE 2.
002762  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                          .MD.=6
                                                                ;/READ FAST PATH REG.
002766  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
002770  117737  176414  001126          MOVB    @MBSEL3,$BDDAT  ;READ FAST PATH RESULT-ANY DATA
002776  001005                          BNE     3$              ;GET XFERRED? YES-NEXT CHECK.
003000  012737  000377  001124          MOV     #377,$GDDAT     ;HAD EXPECTED 377, BUT WOULD HAVE
                                                                ;SETTLED FOR ANYTHING BUT ZERO.

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

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

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

003010  000436                          BR      TST10           ;;

003012                          3$:
                                                                ;/-MCMD-
003012  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003016  001424                          .WORD   KMCSR
        000001                          .MD.=1
                                                                ;/COMMAND=NOP.
003020  000001                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
003022  117737  176416  001126          MOVB    @BSEL5,$BDDAT   ;READ SIDE 1 UBSR.
003030  012737  000052  001124          MOV     #52,$GDDAT      ;EXPECT NORMAL UBSR SETTING.
003036  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BIT4 CLEAR?
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                          4$:
   833  003052  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
   834          000001                          .MD.=1
   835                                                                  ;/COMMAND=NOP.
   836  003056  000001                          .WORD   .MD.
   837                                          ;/RETURN HERE AFTER COMMAND .
   838  003060  117737  176330  001126          MOVB    @MBSEL5,$BDDAT  ;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     $GDDAT,$BDDAT   ;DID BITS SET?
   842  003102  001401                          BEQ     TST10           ;;

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

   846  003104  104000                          ERROR                   ;TESTING THAT SIDE 2 IOSR BIT5
   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                                                                  ;/-MCMD-
   858  003110  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
   859  003114  001424                          .WORD   KMCSR
   860          000006                          .MD.=6
   861                                                                  ;/READ FAST PATH REG.
   862  003116  000006                          .WORD   .MD.
   863                                          ;/RETURN HERE AFTER COMMAND .
   864
   865  003120  112777  000377  176262          MOVB    #377,@MBSEL3    ;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
   869                                                                  ;/WRITE FAST PATH REG.
   870  003132  000007                          .WORD   .MD.
   871                                          ;/RETURN HERE AFTER COMMAND .
   872                                                                  ;ALLOW SETTLE TIME
   873  003134  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
   874          000001                          .MD.=1
   875                                                                  ;/COMMAND=NOP.
```

```
                                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                        MOVB    @MBSEL5,$BDDAT  ;READ SIDE #2 IOSR.
                                        MOV     #76,$GDDAT      ;NORMAL=56, EXPECT BIT 4
                                                                ;TO BE SET.
                                        CMP     $BDDAT,$GDDAT   ;IOSR SIDE 2 OK? (BIT4 SET).
                                        BEQ     1$

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

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

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

                                        BR      TST11           ;;
                                        MOVB    @BSEL5,$BDDAT   ;READ SIDE #1 UBSR
                                        MOV     #12,$GDDAT      ;NORMAL=51,EXPECT BIT5
                                                                ;TO BE CLEAR.
                                        CMP     $GDDAT,$BDDAT   ;IS IT CLEAR?
                                        BEQ     2$              ;YES-NEXT CHECK.

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

                                        ERROR   1               ;BIT5 OF SIDE 1 UBSR NOT CLEAR WHEN
                                                                ;SIDE 2 MADE XFER VIA FAST PATH.

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

                                        BR      TST11           ;;
                                                                ;READ FAST PATH SIDE 1.


                                2$:
                                                                ;/-MCMD-
                                        JSR     R5,ISSUEC       ;/ISSUE COMMAND TO kMC #1.
                                        .WORD   KMCSR
                                        .MD.=6
                                                                ;/READ FAST PATH REG.
                                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                        MOVB    @BSEL3,$BDDAT   ;READ FAST PATH. ANY DATA GET
                                        BNE     3$              ;XFERRED? YES-NEXT CHECK.
                                        MOV     #377,$GDDAT     ;HAD EXPECTED 377, BUT WOULD HAVE
                                                                ;SETTLED FOR ANYTHING BUT ZERO.

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

                                        ERROR   4               ;NO DATA GOT THROUGH FAST PATH
                                                                ;SIDE 2 TO SIDE 1.

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

```
003252  000432                          BR      TST11           ;;
003254                          3$:
003254  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                          .MD.=1
                                                                ;/COMMAND=NOP.
003260  000001                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
003262  117737  176126  001126          MOVB    @MBSEL5,$BDDAT  ;READ SIDE 2 IOSR.
003270  012737  000056  001124          MOV     #56,$GDDAT      ;EXPECT NORMAL IOSR SETTING
003276  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BIT4 CLEAR?
003304  001402                          BEQ     4$              ;YES-NEXT CHECK.

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

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

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

003310  000413                          BR      TST11           ;;

003312  117737  176126  001126  4$:     MOVB    @BSEL5,$BDDAT   ;READ SIDE 1 UBSR.
003320  012737  000052  001124          MOV     #52,$GDDAT      ;EXPECT BITS TO SET AFTER FAST PATH
                                                                ;REG. WAS READ.
003326  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;DID BITS SET?
003334  001401                          BEQ     TST11           ;;

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

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

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




                                ;;*************************************************************
                                ;*TEST 11        *TESt THAT PATTERN 1 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
                                        ;*
                                        ;*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.
                                        ;*
                                ;;*************************************************************
003340  000004                  TST11:  SCOPE
                                                                ;/-DXFRF-
                                                                ;/-MCMD-
003342  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003346  001424                          .WORD   KMCSR
        000001                          .MD.=1
                                                                ;/COMMAND=NOP.
```

```
                                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                        MOV     #1,$GDDAT       ;/RECORD XFERR PATTERN
003352  012737  000001  001124          MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
                                                                ;/-MCMD-
003366  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003372  001424                          .WORD   KMCSR
        000007                          .MD.=7
                                                                ;/WRITE FAST PATH REG.
003374  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
003376  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                          .MD.=6
                                                                ;/READ FAST PATH REG.
003402  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

003404  117737  176000  001126          MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
003412  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
003420  001401                          BEQ     1$

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

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

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

003424  005037  001124          1$:    CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
003430  105077  176004                  CLRB    @BSEL3
                                                                ;/-MCMD-
003434  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003440  001424                          .WORD   KMCSR
        000007                          .MD.=7
                                                                ;/WRITE FAST PATH REG.
003442  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND
003444  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                          .MD.=6
                                                                ;/READ FAST PATH REG.
003450  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

003452  117737  175732  001126          MOVB    @MBSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
003460  001401                          BEQ     TST12           ;;

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

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

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

```
                                ;;*****************************************************************
                                ;*TEST 12        *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
                                ;*
                                ;*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.
                                ;*
                                ;;*****************************************************************
003464  000004                TST12:  SCOPE
                                                        ;/-DXFRF-
                                                        ;/-MCMD-
003466  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003472  001424                        .WORD   KMCSR
        000001                        .MD.=1
                                                        ;/COMMAND=NOP.
003474  000001                        .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

003476  012737  000002  001124        MOV     #2,$GDDAT       ;/RECORD XFERR PATTERN
003504  113777  001124  175726        MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
                                                        ;/-MCMD-
003512  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003516  001424                        .WORD   KMCSR
        000007                        .MD.=7
                                                        ;/WRITE FAST PATH REG.
003520  000007                        .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
003522  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                        .MD.=6
                                                        ;/READ FAST PATH REG.
003526  000006                        .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

003530  117737  175654  001126        MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
003536  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
003544  001401                        BEQ     1$

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

003546  104004                        ERROR   4               ;/IN PROPER XFER OF PATTERN 2 FROM
                                                                ;/SIDE 1 TO SIDE 2

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

003550  005037  001124        1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
003554  105077  175660                CLRB    @BSEL3
                                                        ;/-MCMD-
003560  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
003564  001424                        .WORD   KMCSR
        000007                        .MD.=7
                                                        ;/WRITE FAST PATH REG.
003566  000007                        .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND
003570  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
```

```
                000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
003574          000006                          .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .

003576  117737  175606  001126                  MOVB    @MBSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
003604  001401                                  BEQ     TST13          ;;

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

003606  104004                                  ERROR   4              ;/FAILED TO XFER ZERO PATTERN AFTER

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


                                        ;;**************************************************************
                                        ;*TEST 13        *TEST THAT PATTERN 4 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH REG
                                                ;*
                                                ;*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.
                                                ;*
                                        ;;**************************************************************
003610  000004                          TST13:  SCOPE
                                                        ;/-DXFRF-
                                                        ;/-MCMD-
003612  004537  011656                          JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
003616  001424                                  .WORD   KMCSR
                000001                          .MD.=1
                                                        ;/COMMAND=NOP.
003620  000001                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .


003622  012737  000004  001124                  MOV     #4,$GDDAT      ;/RECORD XFERR PATTERN
003630  113777  001124  175602                  MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
                                                        ;/-MCMD-
003636  004537  011656                          JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
003642  001424                                  .WORD   KMCSR
                000007                          .MD.=7
                                                        ;/WRITE FAST PATH REG.
003644  000007                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
003646  004537  012100                          JSR     R5,KMSIM       ;/DMC SIMULATE KMC INSTR. LIST FOR.
                000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
003652  000006                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .


003654  117737  175530  001126                  MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
003662  023737  001124  001126                  CMP     $GDDAT,$BDDAT  ;/DATA SENT=DATA RECEIVED?
003670  001401                                  BEQ     1$

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

```
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     $GDDAT          ;/NOW XFER A ZERO PATTERN
1152  003700  105077  175534                  CLRB    @BSEL3
1153                                                                  ;/-MCMD-
1154  003704  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1155  003710  001424                          .WORD   KMCSR
1156          000007                          .MD.=7
1157                                                                  ;/WRITE FAST PATH REG.
1158  003712  000007                          .WORD   .MD.
1159                                          ;/RETURN HERE AFTER COMMAND .
1160  003714  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1161          000006                          .MD.=6
1162                                                                  ;/READ FAST PATH REG.
1163  003720  000006                          .WORD   .MD.
1164                                          ;/RETURN HERE AFTER COMMAND .
1165
1166  003722  117737  175462  001126          MOVB    @MBSEL3,$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                                  ;*TEST 14        *TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH RE
1178                                  ;*
1179                                  ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1180                                  ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1181                                  ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1182                                  ;*
1183                                  ;;**********************************************************************
1184  003734  000004                  TST14:  SCOPE
1185                                                                  ;/-DXFRF-
1186                                                                  ;/-MCMD-
1187  003736  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1188  003742  001424                          .WORD   KMCSR
1189          000001                          .MD.=1
1190                                                                  ;/COMMAND=NOP.
1191  003744  000001                          .WORD   .MD.
1192                                          ;/RETURN HERE AFTER COMMAND .
1193
1194  003746  012737  000010  001124          MOV     #10,$GDDAT      ;/RECORD XFERR PATTERN
1195  003754  113777  001124  175456          MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
1196                                                                  ;/-MCMD-
1197  003762  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO kMC #1.
1198  003766  001424                          .WORD   KMCSR
1199          000007                          .MD.=7
```

```
                                                        ;/WRITE FAST PATH REG.
003770  000007                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
003772  004537  012100                          JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                                  .MD.=6
                                                        ;/READ FAST PATH REG.
003776  000006                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .

004000  117737  175404  001126                  MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
004006  023737  001124  001126                  CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
004014  001401                                  BEQ     1$

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

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

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

004020  005037  001124          1$:            CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
004024  105077  175410                          CLRB    @BSEL3
                                                        ;/-MCMD-
004030  004537  011656                          JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
004034  001424                                  .WORD   KMCSR
        000007                                  .MD.=7
                                                        ;/WRITE FAST PATH REG.
004036  000007                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
004040  004537  012100                          JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000006                                  .MD.=6
                                                        ;/READ FAST PATH REG.
004044  000006                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .

004046  117737  175336  001126                  MOVB    @MBSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
004054  001401                                  BEQ     TST15           ;;

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

004056  104004                                  ERROR   4               ;/FAILED TO xFER ZERO PATTERN AFTER

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


                                ;;**************************************************************
                                ;*TEST 15        *TESt THAT PATTERN 20 CAN BE SENT FROM SIDE 1 TO sIDE 2 VIA FAST PATH RE
                                                ;*
                                                ;*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.
                                                ;*
                                ;;**************************************************************
004060  000004                  TST15:  SCOPE
```

```
1254                                                        ;/-DXFRF-
1255                                                        ;/-MCMD-
1256  004062  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1257  004066  001424                        .WORD   KMCSR
1258          000001                        .MD.=1
1259                                                        ;/COMMAND=NOP.
1260  004070  000001                        .WORD   .MD.
1261                                        ;/RETURN HERE AFTER COMMAND .
1262
1263  004072  012737  000020  001124        MOV     #20,$GDDAT      ;/RECORD XFERR PATTERN
1264  004100  113777  001124  175332        MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO U CODE.
1265                                                        ;/-MCMD-
1266  004106  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1267  004112  001424                        .WORD   KMCSR
1268          000007                        .MD.=7
1269                                                        ;/WRITE FAST PATH REG.
1270  004114  000007                        .WORD   .MD.
1271                                        ;/RETURN HERE AFTER COMMAND .
1272  004116  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1273          000006                        .MD.=6
1274                                                        ;/READ FAST PATH REG.
1275  004122  000006                        .WORD   .MD.
1276                                        ;/RETURN HERE AFTER COMMAND .
1277
1278  004124  117737  175260  001126        MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
1279  004132  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
1280  004140  001401                        BEQ     1$

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

1284  004142  104004                        ERROR   4               ;/IN PROPER XFER OF PATTERN 20 FROM
1285                                                        ;/SIDE 1 TO SIDE 2

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

1289  004144  005037  001124        1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
1290  004150  105077  175264                CLRB    @BSEL3
1291                                                        ;/-MCMD-
1292  004154  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1293  004160  001424                        .WORD   KMCSR
1294          000007                        .MD.=7
1295                                                        ;/WRITE FAST PATH REG.
1296  004162  000007                        .WORD   .MD.
1297                                        ;/RETURN HERE AFTER COMMAND .
1298  004164  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1299          000006                        .MD.=6
1300                                                        ;/READ FAST PATH REG.
1301  004170  000006                        .WORD   .MD.
1302                                        ;/RETURN HERE AFTER COMMAND .
1303
1304  004172  117737  175212  001126        MOVB    @MBSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1305  004200  001401                        BEQ     TST16           ;;

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

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

1313
1314                                  ;;*****************************************************************
1315                                  ;*TEST 16       *TESt THAT PATTERN 40 CAN BE SENT FROM SIDE 1 TO sIDE 2 VIA FAST PATH RE
1316                                          ;*
1317                                          ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 1
1318                                          ;*TO SIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1319                                          ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1320                                          ;*
1321                                  ;;*****************************************************************
1322  004204  000004                  TST16:  SCOPE
1323                                                                  ;/-DXFRF-
1324                                                                  ;/-MCMD-
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    $GDDAT,@BSEL3 ;/WRITE PATTERN TO u CODE.
1334                                                                  ;/-MCMD-
1335  004232  004537  011656                  JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1336  004236  001424                          .WORD   KMCSR
1337           000007                         .MD.=7
1338                                                                  ;/WRITE FAST PATH REG.
1339  004240  000007                          .WORD   .MD.
1340                                          ;/RETURN HERE AFTER COMMAND .
1341  004242  004537  012100                  JSR     R5,KMSIM       ;/DMC SIMULATE KMC INSTR. LIST FOR.
1342           000006                         .MD.=6
1343                                                                  ;/READ FAST PATH REG.
1344  004246  000006                          .WORD   .MD.
1345                                          ;/RETURN HERE AFTER COMMAND .
1346
1347  004250  117737  175134  001126          MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
1348  004256  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;/DATA SENT=DATA RECEIVED?
1349  004264  001401                          BEQ     1$
                                              ;**********>> ERROR <<**********

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

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

```
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    @MBSEL3,$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  004332  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1395  004336  001424                          .WORD   KMCSR
1396          000001                          .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          MOVB    @MBSEL3,$BDDAT ;/READ SIDE #2
1417  004402  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;/DATA SENT=DATA RECEIVED?
1418  004410  001401                          BEQ     1$

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

1422  004412  104004                          ERROR   4              ;/IN PROPER XFER OF PATTERN 100 FROM
1423                                                                 ;/SIDE 1 TO SIDE 2

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

1427  004414  005037  001124          1$:     CLR     $GDDAT         ;/NOW XFER A ZERO PATTERN
1428  004420  105077  175014                  CLRB    @BSEL3
1429                                                                 ;/-MCMD-
1430  004424  004537  011656                  JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1431  004430  001424                          .WORD   KMCSR
1432          000007                          .MD.=7
1433                                                                 ;/WRITE FAST PATH REG.
1434  004432  000007                          .WORD   .MD.
1435                                          ;/RETURN HERE AFTER COMMAND .
1436  004434  004537  012100                  JSR     R5,KMSIM       ;/DMC SIMULATE KMC INSTR. LIST FOR.
1437          000006                          .MD.=6
1438                                                                 ;/READ FAST PATH REG.
1439  004440  000006                          .WORD   .MD.
1440                                          ;/RETURN HERE AFTER COMMAND .
1441
1442  004442  117737  174742  001126          MOVB    @MBSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1443  004450  001401                          BEQ     TST20          ;;

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

1447  004452  104004                          ERROR   4              ;/FAILED TO XFER ZERO PATTERN AFTER

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

1451
1452                                  ;;=====================================================================
1453                                  ;*TEST 20       *TEST THAT PATTERN 200 CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA FAST PATH R
1454                                          ;*
1455                                          ;*IN THIS TEST WE ARE GOING TO xFER DATA FROM SIDE 1
1456                                          ;*TO sIDE 2 AND CHECK THE RESULTS. WE KNOW FROM A
1457                                          ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1458                                          ;*
1459                                  ;;=====================================================================
1460  004454  000004                  TST20:  SCOPE
1461                                                                 ;/-DXFRF-
1462                                                                 ;/-MCMD-
1463  004456  004537  011656                  JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
1464  004462  001424                          .WORD   KMCSR
1465          000001                          .MD.=1
1466                                                                 ;/COMMAND=NOP.
1467  004464  000001                          .WORD   .MD.
1468                                          ;/RETURN HERE AFTER COMMAND .
1469
```

```
1470  004466  012737  000200  001124          MOV     #200,$GDDAT     ;/RECORD XFERR PATTERN
1471  004474  113777  001124  174736          MOVB    $GDDAT,@BSEL3 ;/WRITE PATTERN TO u CODE.
1472                                                                  ;/-MCMD-
1473  004502  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1474  004506  001424                          .WORD   KMCSR
1475          000007                          .MD.=7
1476                                                                  ;/WRITE FAST PATH REG.
1477  004510  000007                          .WORD   .MD.
1478                                          ;/RETURN HERE AFTER COMMAND .
1479  004512  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1480          000006                          .MD.=6
1481                                                                  ;/READ FAST PATH REG.
1482  004516  000006                          .WORD   .MD.
1483                                          ;/RETURN HERE AFTER COMMAND .
1484
1485  004520  117737  174664  001126          MOVB    @MBSEL3,$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     $GDDAT          ;/NOW XFER A ZERO PATTERN
1497  004544  105077  174670                  CLRB    @BSEL3
1498                                                                  ;/-MCMD-
1499  004550  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1500  004554  001424                          .WORD   KMCSR
1501          000007                          .MD.=7
1502                                                                  ;/WRITE FAST PATH REG.
1503  004556  000007                          .WORD   .MD.
1504                                          ;/RETURN HERE AFTER COMMAND .
1505  004560  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1506          000006                          .MD.=6
1507                                                                  ;/READ FAST PATH REG.
1508  004564  000006                          .WORD   .MD.
1509                                          ;/RETURN HERE AFTER COMMAND .
1510
1511  004566  117737  174616  001126          MOVB    @MBSEL3,$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                                  ;*TEST 21        *TEST THAT PATTERN 1 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1523                                  ;*
```

```
                                        ;*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.
                                        ;*
                                ;;**************************************************************
004600  000004                  TST21:  SCOPE
                                                        ;/-DXFRF-
004602  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                          .MD.=1
                                                        ;/COMMAND=NOP.
004606  000001                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

004610  012737  000001  001124          MOV     #1,$GDDAT       ;/RECORD XFERR PATTERN
004616  113777  001124  174564          MOVB    $GDDAT,@MBSEL3  ;/WRITE PATTERN TO U CODE.
004624  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                          .MD.=7
                                                        ;/WRITE FAST PATH REG.
004630  000007                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
004632  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
004636  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
004640  000006                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

004642  117737  174572  001126          MOVB    @BSEL3,$BDDAT   ;/READ SIDE #1
004650  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
004656  001401                          BEQ     1$

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

004660  104003                          ERROR   3               ;INPROPER XFER OF PATTERN 1 FROM
                                                                ;/SIDE 2 TO sIDE 1

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

004662  005037  001124          1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
004666  105077  174516                  CLRB    @MBSEL3
004672  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                          .MD.=7
                                                        ;/WRITE FAST PATH REG.
004676  000007                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
004700  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
004704  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
004706  000006                  .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
```

```
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                                  ;*TEST 22        *TEST THAT PATTERN 2 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH REG
1590                                          ;*
1591                                          ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
1592                                          ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
1593                                          ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1594                                          ;*
1595                                  ;;*****************************************************************
1596  004722  000004                  TST22:  SCOPE
1597                                                                ;/-DXFRF-
1598  004724  004537  012100                  JSR     R5,KMSIM     ;/DMC SIMULATE KMC INSTR. LIST FOR.
1599           000001                          .MD.=1
1600                                                                ;/COMMAND=NOP.
1601  004730  000001                          .WORD   .MD.
1602                                          ;/RETURN HERE AFTER COMMAND .
1603
1604  004732  012737  000002  001124          MOV     #2,$GDDAT    ;/RECORD XFERR PATTERN
1605  004740  113777  001124  174442          MOVB    $GDDAT,@MBSEL3 ;/WRITE PATTERN TO U CODE.
1606  004746  004537  012100                  JSR     R5,KMSIM     ;/DMC SIMULATE KMC INSTR. LIST FOR.
1607           000007                          .MD.=7
1608                                                                ;/WRITE FAST PATH REG.
1609  004752  000007                          .WORD   .MD.
1610                                          ;/RETURN HERE AFTER COMMAND .
1611                                                                ;/-MCMD-
1612  004754  004537  011656                  JSR     R5,ISSUEC    ;/ISSUE COMMAND TO KMC #1.
1613  004760  001424                          .WORD   KMCSR
1614           000006                          .MD.=6
1615                                                                ;/READ FAST PATH REG.
1616  004762  000006                          .WORD   .MD.
1617                                          ;/RETURN HERE AFTER COMMAND .
1618
1619  004764  117737  174450  001126          MOVB    @BSEL3,$BDDAT ;/READ SIDE #1
1620  004772  023737  001124  001126          CMP     $GDDAT,$BDDAT  ;/DATA SENT=DATA RECEIVED?
1621  005000  001401                          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          1$:     CLR     $GDDAT       ;/NOW XFER A ZERO PATTERN
1631  005010  105077  174374                  CLRB    @MBSEL3
```

```
1632  005014  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1633          000007                          .MD.=7
1634                                          ;/WRITE FAST PATH REG.
1635  005020  000007                          .WORD   .MD.
1636                                          ;/RETURN HERE AFTER COMMAND .
1637                                          ;/-MCMD-
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
1664                                          ;/-DXFRF-
1665  005046  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
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,@MBSEL3 ;/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                                          ;/-MCMD-
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          MOVB    @BSEL3,$BDDAT ;/READ SIDE #1
1687  005114  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
1688  005122  001401                          BEQ     1$

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

1692  005124  104003                          ERROR   3               ;INPROPER XFER OF PATTERN 4 FROM
1693                                                                  ;/SIDE 2 TO SIDE 1

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

1697  005126  005037  001124          1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
1698  005132  105077  174252                  CLRB    @MBSEL3
1699  005136  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1700            000007                        .MD.=7
1701                                                                  ;/WRITE FAST PATH REG.
1702  005142  000007                          .WORD   .MD.
1703                                          ;/RETURN HERE AFTER COMMAND .
1704                                                                  ;/-MCMD-
1705  005144  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1706  005150  001424                          .WORD   KMCSR
1707            000006                        .MD.=6
1708                                                                  ;/READ FAST PATH REG.
1709  005152  000006                          .WORD   .MD.
1710                                          ;/RETURN HERE AFTER COMMAND .
1711
1712  005154  117737  174260  001126          MOVB    @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1713  005162  001401                          BEQ     TST24           ;;

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

1717  005164  104003                          ERROR   3               ;FAILED TO XFER ZERO PATTERN.

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

1721
1722                                  ;;**************************************************************************
1723                                  ;*TEST 24        *TEST THAT PATTERN 10 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH RE
1724                                          ;*
1725                                          ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
1726                                          ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
1727                                          ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
1728                                          ;*
1729                                  ;;**************************************************************************
1730  005166  000004                  TST24:  SCOPE
1731                                                                  ;/-DXFRF-
1732  005170  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1733            000001                        .MD.=1
1734                                                                  ;/COMMAND=NOP.
1735  005174  000001                          .WORD   .MD.
1736                                          ;/RETURN HERE AFTER COMMAND .
1737
1738  005176  012737  000010  001124          MOV     #10,$GDDAT      ;/RECORD XFERR PATTERN
1739  005204  113777  001124  174176          MOVB    $GDDAT,@MBSEL3 ;/WRITE PATTERN TO U CODE.
```

```
1740  005212  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1741          000007                        .MD.=7
1742                                        ;/WRITE FAST PATH REG.
1743  005216  000007                        .WORD   .MD.
1744                                        ;/RETURN HERE AFTER COMMAND .
1745                                        ;/-MCMD-
1746  005220  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1747  005224  001424                        .WORD   KMCSR
1748          000006                        .MD.=6
1749                                        ;/READ FAST PATH REG.
1750  005226  000006                        .WORD   .MD.
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     1$

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

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

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

1764  005250  005037  001124        1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
1765  005254  105077  174130                CLRB    @MBSEL3
1766  005260  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1767          000007                        .MD.=7
1768                                        ;/WRITE FAST PATH REG.
1769  005264  000007                        .WORD   .MD.
1770                                        ;/RETURN HERE AFTER COMMAND .
1771                                        ;/-MCMD-
1772  005266  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1773  005272  001424                        .WORD   KMCSR
1774          000006                        .MD.=6
1775                                        ;/READ FAST PATH REG.
1776  005274  000006                        .WORD   .MD.
1777                                        ;/RETURN HERE AFTER COMMAND .
1778
1779  005276  117737  174136  001126        MOVB    @BSEL3,$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
```

```
                                        ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
                                        ;*
                                ;;*****************************************************************
005310  000004                 TST25:  SCOPE
                                                        ;/-DXFRF-
005312  004537  012100                 JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                  .MD.=1
                                                        ;/COMMAND=NOP.
005316  000001                 .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

005320  012737  000020  001124  MOV     #20,$GDDAT      ;/RECORD XFERR PATTERN
005326  113777  001124  174054  MOVB    $GDDAT,@MBSEL3  ;/WRITE PATTERN TO U CODE.
005334  004537  012100          JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                  .MD.=7
                                                        ;/WRITE FAST PATH REG.
005340  000007                 .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005342  004537  011656          JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
005346  001424                  .WORD   KMCSR
        000006                  .MD.=6
                                                        ;/READ FAST PATH REG.
005350  000006                 .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

005352  117737  174062  001126  MOVB    @BSEL3,$BDDAT   ;/READ SIDE #1
005360  023737  001124  001126  CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
005366  001401                  BEQ     1$

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

005370  104003                 ERROR   3               ;IMPROPER XFER OF PATTERN 20 FROM
                                                        ;/SIDE 2 TO SIDE 1

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

005372  005037  001124  1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
005376  105077  174006          CLRB    @MBSEL3
005402  004537  012100          JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                  .MD.=7
                                                        ;/WRITE FAST PATH REG.
005406  000007                 .WORd   .MD.
                                ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005410  004537  011656          JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
005414  001424                  .WORD   KMCSR
        000006                  .MD.=6
                                                        ;/READ FAST PATH REG.
005416  000006                 .WORD   .MD.
                                ;/RETURN HERE AFTER COMMAND .

005420  117737  174014  001126  MOVB    @BSEL3,$BDDAT   ;/READ PATTERN ERROR IF NON-ZERO
005426  001401                  BEQ     TST26           ;;
```

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

 1851  005430  104003                   ERROR   3               ;FAILED TO XFER ZERO PATTERN.

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

 1855
 1856                                ;;*****************************************************************
 1857                                ;*TEST 26        *TEST THAT PATTERN 40 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH RE
 1858                                        ;*
 1859                                        ;*IN THIS TEST WE ARE GOING TO XFER DATA FROM SIDE 2
 1860                                        ;*TO SIDE 1 AND CHECK THE RESULTS. WE KNOW FROM A
 1861                                        ;*PREVIOUS TEST THAT SOME DATA CAN GET THROUGH.
 1862                                        ;*
 1863                                ;;*****************************************************************
 1864  005432  000004                TST26:  SCOPE
 1865                                                                ;/-DXFRF-
 1866  005434  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
 1867           000001                       .MD.=1
 1868                                                                ;/COMMAND=NOP.
 1869  005440  000001                       .WORD   .MD.
 1870                                        ;/RETURN HERE AFTER COMMAND .
 1871
 1872  005442  012737  000040  001124        MOV     #40,$GDDAT      ;/RECORD XFERR PATTERN
 1873  005450  113777  001124  173732        MOVB    $GDDAT,@MBSEL3  ;/WRITE PATTERN TO U CODE.
 1874  005456  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
 1875           000007                       .MD.=7
 1876                                                                ;/WRITE FAST PATH REG.
 1877  005462  000007                       .WORD   .MD.
 1878                                        ;/RETURN HERE AFTER COMMAND .
 1879                                                                ;/-MCMD-
 1880  005464  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
 1881  005470  001424                       .WORD   KMCSR
 1882           000006                       .MD.=6
 1883                                                                ;/READ FAST PATH REG.
 1884  005472  000006                       .WORD   .MD.
 1885                                        ;/RETURN HERE AFTER COMMAND .
 1886
 1887  005474  117737  173740  001126        MOVB    @BSEL3,$BDDAT   ;/READ SIDE #1
 1888  005502  023737  001124  001126        CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
 1889  005510  001401                       BEQ     1$

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

 1893  005512  104003                   ERROR   3               ;INPROPER XFER OF PATTERN 40 FROM
 1894                                                                ;/SIDE 2 TO SIDE 1

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

 1898  005514  005037  001124        1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
 1899  005520  105077  173664                CLRB    @MBSEL3
 1900  005524  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
 1901           000007                       .MD.=7
```

```
                                                        ;/WRITE FAST PATH REG.
005530  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005532  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
005536  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
005540  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

005542  117737  173672  001126          MOVB    @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
005550  001401                          BEQ     TST27           ;;

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

005552  104003                          ERROR   3               ;FAILED TO xFER ZERO PATTERN.

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


                                ;;**************************************************************
                                ;*TEST 27        *TEST THAT PATTERN 100 CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA FAST PATH R
                                ;*
                                ;*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.
                                ;*
                                ;;**************************************************************
005554  000004                  TST27:  SCOPE
                                                        ;/-DXFRF-
005556  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                          .MD.=1
                                                        ;/COMMAND=NOP.
005562  000001                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

005564  012737  000100  001124          MOV     #100,$GDDAT     ;/RECORD XFERR PATTERN
005572  113777  001124  173610          MOVB    $GDDAT,@MBSEL3 ;/WRITE PATTERN TO U CODE.
005600  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                          .MD.=7
                                                        ;/WRITE FAST PATH REG.
005604  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005606  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
005612  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
005614  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

005616  117737  173616  001126          MOVB    @BSEL3,$BDDAT ;/READ SIDE #1
005624  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
```

```
1956  005632  001401                          BEQ     1$

                                              ;**********>> EPROR <<**********

1960  005634  104003                          ERROR   3               ;INPROPER XFER OF PATTERN 100 FROM
1961                                                                  ;/SIDE 2 TO SIDE 1

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

1965  005636  005037  001124          1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
1966  005642  105077  173542                  CLRB    @MBSEL3
1967  005646  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
1968          000007                          .MD.=7
1969                                                                  ;/WRITE FAST PATH REG.
1970  005652  000007                          .WORD   .MD.
1971                                          ;/RETURN HERE AFTER COMMAND .
1972                                                                  ;/-MCMD-
1973  005654  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
1974  005660  001424                          .WORD   KMCSR
1975          000006                          .MD.=6
1976                                                                  ;/READ FAST PATH REG.
1977  005662  000006                          .WORD   .MD.
1978                                          ;/RETURN HERE AFTER COMMAND .
1979
1980  005664  117737  173550  001126          MOVB    @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
1981  005672  001401                          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                  TST30:  SCOPE
1999                                                                  ;/-DXFRF-
2000  005700  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2001          000001                          .MD.=1
2002                                                                  ;/COMMAND=NOP.
2003  005704  000001                          .WORD   .MD.
2004                                          ;/RETURN HERE AFTER COMMAND .
2005
2006  005706  012737  000200  001124          MOV     #200,$GDDAT     ;/RECORD XFERR PATTERN
2007  005714  113777  001124  173466          MOVB    $GDDAT,@MBSEL3 ;/WRITE PATTERN TO u CODE.
2008  005722  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2009          000007                          .MD.=7
```

```
                                                        ;/WRITE FAST PATH REG.
005726  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005730  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
005734  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
005736  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

005740  117737  173474  001126          MOVB    @BSEL3,$BDDAT ;/READ SIDE #1
005746  023737  001124  001126          CMP     $GDDAT,$BDDAT   ;/DATA SENT=DATA RECEIVED?
005754  001401                          BEQ     1$

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

005756  104003                          ERROR   3               ;INPROPER XFER OF PATTERN 200 FROM
                                                                ;/SIDE 2 TO SIDE 1

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

005760  005037  001124          1$:     CLR     $GDDAT          ;/NOW XFER A ZERO PATTERN
005764  105077  173420                  CLRB    @MBSEL3
005770  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000007                          .MD.=7
                                                        ;/WRITE FAST PATH REG.
005774  000007                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                        ;/-MCMD-
005776  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
006002  001424                          .WORD   KMCSR
        000006                          .MD.=6
                                                        ;/READ FAST PATH REG.
006004  000006                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

006006  117737  173426  001126          MOVB    @BSEL3,$BDDAT ;/READ PATTERN ERROR IF NON-ZERO
006014  001401                          BEQ     TST31           ;;

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

006016  104003                          ERROR   3               ;FAILED TO XFER ZERO PATTERN.

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


                                ;;**************************************************************
                                ;*TEST 31        *TEST THAT DATA CAN BE SENT FROM SIDE 1 TO SIDE 2 VIA SILO
                                ;;**************************************************************
006020  000004                  TST31:  SCOPE

006022  004737  012524                  JSR     PC,BINT         ;INIT. KMCS.
006026  112777  000377  173404          MOVB    #377,@BSEL3     ;DATA TO BE XFERRED.
```

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

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

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

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

006100  000413                        BR      TST32           ;;

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

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

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

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


                                ;;*****************************************************************
                                ;*TEST 32        *TEST THAT DATA CAN BE SENT FROM SIDE 2 TO SIDE 1 VIA SILO
                                ;;*****************************************************************
006130  000004                TST32:  SCOPE

006132  004737  012524                JSR     PC,BINT         ;INIT KMCS.
```

```
006136  112777  000377  173244               MOVB    #377,@MBSEL3      ;DATA TO BE XFERRED.
006144  004537  012100                       JSR     R5,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                               .MD.=5
                                                                       ;/WRITE TO SILO.
006150  000005                               .WORD   .MD.
                                             ;/RETURN HERE AFTER COMMAND .
006152  004537  012100                       JSR     R5,KMSIM          ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                               .MD.=1
                                                                       ;/COMMAND=NOP.
006156  000001                               .WORd   .MD.
                                             ;/RETURN HERE AFTER COMMAND .
006160  117737  173260  001126               MOVB    @BSEL5,$BDDAT     ;READ SIDE 1 UBSR
006166  012737  000050  001124               MOV     #50,$GDDAT        ;EXPECT BIT0 TO BE CLEAR.
006174  023737  001124  001126               CMP     $GDDAT,$BDDAT     ;DIC BIT0 CLEAR?
006202  001402                               BEQ     1$

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

006204  104001                               ERROR   1                 ;BIT 1 UBSR SIDE 1 FAILED
                                                                       ;TO CLEAR (INDICATION SILO DATA
                                                                       ;PRESENT) WHEN DATA WAS
                                                                       ;WRITTEN INTO SIDE 2 SILO

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

006206  000414                               BR      TST33             ;;
006210                               1$:
                                                                       ;/-MCMD-
006210  004537  011656                       JSR     R5,ISSUEC         ;/ISSUE COMMAND TO KMC #1.
006214  001424                               .WORD   KMCSR
        000004                               .MD.=4
                                                                       ;/READ SILO.
006216  000004                               .WORD   .MD.
                                             ;/RETURN HERE AFTER COMMAND .
006220  117737  173214  001126               MOVB    @BSEL3,$BDDAT     ;READ DATA.
006226  001004                               BNE     TST33             ;;

006230  012737  000377  001124               MOV     #377,$GDDAT       ;HAD EXPECTED 377 BUT WOULD
                                                                       ;HAVE SETTLED FOR ANYTHING TO
                                                                       ;INDICATE SOME DATA GOT THROUGH.

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

006236  104005                               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 XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
                                     ;;*************************************************************
006240  000004                       TST33:  SCOPE
                                                                       ;/-SIDA-
006242  012737  000000  001124               MOV     #0,$GDDAT         ;/RECORD XFER 0
006250  113777  001124  173162               MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
```

```
                                                        ;/-MCMD-
006256  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
006262  001424                        .WORD   KMCSR
        000005                        .MD.=5
                                                        ;/WRITE TO SILO.
006264  000005                        .WORD   .MD.
                                      ;/RETURN HERE AFTER COMMAND .
006266  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000004                        .MD.=4
                                                        ;/READ SILO.
006272  000004                        .WORD   .MD.
                                      ;/RETURN HERE AFTER COMMAND .

006274  117737  173110  001126        MOVB    @MBSEL3,$BDDAT ;/READ FROM REG.
006302  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
006310  001401                        BEQ     TST34           ;;

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

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

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


                                ;;**********************************************************************
                                ;*TEST 34        *TEST THAT DATA PATTERN 1 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
                                ;;**********************************************************************
006314  000004                TST34:  SCOPE
                                                        ;/-SIDA-
006316  012737  000001  001124        MOV     #1,$GDDAT       ;/RECORD XFER 1
006324  113777  001124  173106        MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
                                                        ;/-MCMD-
006332  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO kMC #1.
006336  001424                        .WORD   KMCSR
        000005                        .MD.=5
                                                        ;/WRITE TO SILO.
006340  000005                        .WORD   .MD.
                                      ;/RETURN HERE AFTER COMMAND .
006342  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000004                        .MD.=4
                                                        ;/READ SILO.
006346  000004                        .WORD   .MD.
                                      ;/RETURN HERE AFTER COMMAND .

006350  117737  173034  001126        MOVB    @MBSEL3,$BDDAT ;/READ FROM REG.
006356  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
006364  001401                        BEQ     TST35           ;;

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

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

```
2226                                                    ;/XFERR ERROR VIA SILO.

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

2230
2231                                    ;;*****************************************************************
2232                                    ;*TEST 35        *TEST THAT DATA PATTERN 2 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2233                                    ;;*****************************************************************
2234  006370  000004                    TST35:  SCOPE
2235                                                    ;/-SIDA-
2236  006372  012737  000002  001124            MOV     #2,$GDDAT       ;/RECORD XFER 2
2237  006400  113777  001124  173032            MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
2238                                                    ;/-MCMD-
2239  006406  004537  011656                    JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2240  006412  001424                            .WORD   KMCSR
2241          000005                            .MD.=5
2242                                                    ;/WRITE TO SILO.
2243  006414  000005                            .WORD   .MD.
2244                                            ;/RETURN HERE AFTER COMMAND
2245  006416  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2246          000004                            .MD.=4
2247                                                    ;/READ SILO.
2248  006422  000004                            .WORD   .MD.
2249                                            ;/RETURN HERE AFTER COMMAND .
2250
2251  006424  117737  172760  001126            MOVB    @MBSEL3,$BDDAT ;/READ FROM REG.
2252  006432  123737  001124  001126            CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
2253  006440  001401                            BEQ     TST36           ;;
2254

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

2258  006442  104006                            ERROR   6               ;/SIDE 1 TO SIDE 2 DATA
2259                                                    ;/XFERR ERROR VIA SILO.

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

2263
2264                                    ;;*****************************************************************
2265                                    ;*TEST 36        *TEST THAT DATA PATTERN 4 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE 2
2266                                    ;;*****************************************************************
2267  006444  000004                    TST36:  SCOPE
2268                                                    ;/-SIDA-
2269  006446  012737  000004  001124            MOV     #4,$GDDAT       ;/RECORD XFER 4
2270  006454  113777  001124  172756            MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
2271                                                    ;/-MCMD-
2272  006462  004537  011656                    JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2273  006466  001424                            .WORD   KMCSR
2274          000005                            .MD.=5
2275                                                    ;/WRITE TO SILO.
2276  006470  000005                            .WORD   .MD.
2277                                            ;/RETURN HERE AFTER COMMAND .
2278  006472  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2279          000004                            .MD.=4
```

```
2280                                                    ;/READ SILO.
2281  006476  000004                    .WORD   .MD.
2282                                    ;/RETURN HERE AFTER COMMAND .
2283
2284  006500  117737  172704  001126    MOVB    @MBSEL3,$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    $GDDAT,@BSEL3 ;/DATA TO XFERR.
2304                                                            ;/-MCMD-
2305  006536  004537  011656            JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2306  006542  001424                    .WORD   KMCSR
2307          000005                    .MD.=5
2308                                                    ;/WRITE TO SILO.
2309  006544  000005                    .WORD   .MD.
2310                                    ;/RETURN HERE AFTER COMMAND .
2311  006546  004537  012100            JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2312          000004                    .MD.=4
2313                                                    ;/READ SILO.
2314  006552  000004                    .WORD   .MD.
2315                                    ;/RETURN HERE AFTER COMMAND .
2316
2317  006554  117737  172630  001126    MOVB    @MBSEL3,$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
```

```
2334
2335  006576  012737  000020  001124          MOV     #20,$GDDAT      ;/-SIDA-
2336  006604  113777  001124  172626          MOVB    $GDDAT,@BSEL3 ;/DATA TO xFERR.
2337                                                                  ;/-MCMD-
2338  006612  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2339  006616  001424                          .WORD   KMCSR
2340          000005                          .MD.=5
2341                                                                  ;/WRITE TO SILO.
2342  006620  000005                          .WORD   .MD.
2343                                          ;/RETURN HERE AFTER COMMAND .
2344  006622  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2345          000004                          .MD.=4
2346                                                                  ;/READ SILO.
2347  006626  000004                          .WORD   .MD.
2348                                          ;/RETURN HERE AFTER COMMAND .
2349
2350  006630  117737  172554  001126          MOVB    @MBSEL3,$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 sIDE 2 DATA
2358                                                                  ;/XFERR ERROR VIA SILO.

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

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

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

2390  006722  104006                    ERROR   6               ;/SIDE 1 TO SIDE 2 DATA
2391                                                            ;/XFERR ERROR VIA SILO.

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


2395
2396                                ;;**************************************************************
2397                                ;*TEST 42       *TEST THAT DATA PATTERN 100 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE
2398                                ;;**************************************************************
2399  006724  000004                TST42:  SCOPE
2400                                                            ;/-SIDA-
2401  006726  012737  000100  001124        MOV     #100,$GDDAT     ;/RECORD XFER 100
2402  006734  113777  001124  172476        MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
2403                                                            ;/-MCMD-
2404  006742  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO kMC #1.
2405  006746  001424                        .WORD   KMCSR
2406          000005                        .MD.=5
2407                                                            ;/WRITE TO SILO.
2408  006750  000005                        .WORD   .MD.
2409                                        ;/RETURN HERE AFTER COMMAND .
2410  006752  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2411          000004                        .MD.=4
2412                                                            ;/READ SILO.
2413  006756  000004                        .WORD   .MD.
2414                                        ;/RETURN HERE AFTER COMMAND .
2415
2416  006760  117737  172424  001126        MOVB    @MBSEL3,$BDDAT ;/READ FROM REG.
2417  006766  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
2418  006774  001401                        BEQ     TST43           ;;
2419

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

2423  006776  104006                    ERROR   6               ;/SIDE 1 TO SIDE 2 DATA
2424                                                            ;/XFERR ERROR VIA SILO.

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


2428
2429                                ;;**************************************************************
2430                                ;*TEST 43       *TEST THAT DATA PATTERN 200 CAN BE XFERRED VIA SILO FROM SIDE 1 TO SIDE
2431                                ;;**************************************************************
2432  007000  000004                TST43:  SCOPE
2433                                                            ;/-SIDA-
2434  007002  012737  000200  001124        MOV     #200,$GDDAT     ;/RECORD XFER 200
2435  007010  113777  001124  172422        MOVB    $GDDAT,@BSEL3 ;/DATA TO XFERR.
2436                                                            ;/-MCMD-
2437  007016  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2438  007022  001424                        .WORD   KMCSR
2439          000005                        .MD.=5
2440                                                            ;/WRITE TO SILO.
2441  007024  000005                        .WORD   .MD.
```

```
                                        ;/RETURN HERE AFTER COMMAND .
007026  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000004                          .MD.=4
                                        ;/READ SILO.
007032  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007034  117737  172350  001126          MOVB    @MBSEL3,$BDDAT ;/READ FROM REG.
007042  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007050  001401                          BEQ     TST44           ;;

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

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

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


                                ;;**************************************************************
                                ;*TEST 44        *TEST THAT DATA PATTERN 0 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
                                ;;**************************************************************
007054  000004                  TST44:  SCOPE
                                                                ;/-SIDA-
007056  012737  000000  001124          MOV     #0,$GDDAT       ;/RECORD XFER 0
007064  113777  001124  172316          MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
007072  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                          .MD.=5
                                                                ;/WRITE TO SILO.
007076  000005                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007100  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007104  001424                          .WORD   KMCSR
        000004                          .MD.=4
                                                                ;/READ SILO.
007106  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007110  117737  172324  001126          MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
007116  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007124  001401                          BEQ     TST45           ;;

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

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

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


                                ;;**************************************************************
```

```
                                ;*TEST 45       *TEST THAT DATA PATTERN 1 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
                                ;;*****************************************************************
007130  000004                  TST45:  SCOPE
                                                                ;/-SIDA-
007132  012737  000001  001124          MOV     #1,$GDDAT       ;/RECORD XFER 1
007140  113777  001124  172242          MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
007146  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                          .MD.=5
                                                                ;/WRITE TO SILO.
007152  000005                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007154  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007160  001424                          .WORD   KMCSR
        000004                          .MD.=4
                                                                ;/READ SILO.
007162  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007164  117737  172250  001126          MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
007172  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007200  001401                          BEQ     TST46           ;;

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

007202  104005                          ERROR   5               ;/SIDE 2 TO SIDE 1 DATA
                                                                ;/XFERR ERROR VIA SILO.

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


                                ;;*****************************************************************
                                ;*TEST 46       *TEST THAT DATA PATTERN 2 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
                                ;;*****************************************************************
007204  000004                  TST46:  SCOPE
                                                                ;/-SIDA-
007206  012737  000002  001124          MOV     #2,$GDDAT       ;/RECORD XFER 2
007214  113777  001124  172166          MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
007222  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                          .MD.=5
                                                                ;/WRITE TO SILO.
007226  000005                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007230  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007234  001424                          .WORD   KMCSR
        000004                          .MD.=4
                                                                ;/READ SILO.
007236  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007240  117737  172174  001126          MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
007246  123737  001124  001126          CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007254  001401                          BEQ     TST47           ;;
```

```
                                        ;**********>> 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 XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2561                                ;;**************************************************************
2562  007260  000004                TST47:  SCOPE
2563                                                            ;/-SIDA-
2564  007262  012737  000004  001124        MOV     #4,$GDDAT       ;/RECORD XFER 4
2565  007270  113777  001124  172112        MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
2566  007276  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2567          000005                        .MD.=5
2568                                                            ;/WRITE TO SILO.
2569  007302  000005                        .WORD   .MD.
2570                                        ;/RETURN HERE AFTER COMMAND .
2571                                                            ;/-MCMD-
2572  007304  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2573  007310  001424                        .WORD   KMCSR
2574          000004                        .MD.=4
2575                                                            ;/READ SILO.
2576  007312  000004                        .WORD   .MD.
2577                                        ;/RETURN HERE AFTER COMMAND .
2578
2579  007314  117737  172120  001126        MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
2580  007322  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
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 XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2593                                ;;**************************************************************
2594  007334  000004                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
2600                                                            ;/WRITE TO SILO.
2601  007356  000005                        .WORD   .MD.
2602                                        ;/RETURN HERE AFTER COMMAND .
2603                                                            ;/-MCMD-
```

```
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                                ;*TEST 51        *TEST THAT DATA PATTERN 20 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE 1
2625                                ;;*****************************************************************
2626  007410  000004                TST51:  SCOPE
2627                                                          ;/-SIDA-
2628  007412  012737  000020  001124  MOV     #20,$GDDAT      ;/RECORD XFER 20
2629  007420  113777  001124  171762  MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
2630  007426  004537  012100          JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2631          000005                  .MD.=5
2632                                                          ;/WRITE TO SILO.
2633  007432  000005                  .WORD   .MD.
2634                                  ;/RETURN HERE AFTER COMMAND .
2635                                                          ;/-MCMD-
2636  007434  004537  011656          JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
2637  007440  001424                  .WORD   KMCSR
2638          000004                  .MD.=4
2639                                                          ;/READ SILO.
2640  007442  000004                  .WORD   .MD.
2641                                  ;/RETURN HERE AFTER COMMAND .
2642
2643  007444  117737  171770  001126  MOVB    @BSEL3,$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                                ;;*****************************************************************
2656                                ;*TEST 52        *TEST THAT DATA PATTERN 40 CAN BE XFERRED VIA SILO FROM SIDE 2 TO sIDE 1
2657                                ;;*****************************************************************
```

```
                                TST52:  SCOPE
                                                                ;/-SIDA-
007466  012737  000040  001124            MOV     #40,$GDDAT      ;/RECORD XFER 40
007474  113777  001124  171706            MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
007502  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                          .MD.=5
                                                                ;/WRITE TO SILO.
007506  000005                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007510  004537  011656                    JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007514  001424                          .WORD   KMCSR
        000004                          .MD.=4
                                                                ;/READ SILO.
007516  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007520  117737  171714  001126            MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
007526  123737  001124  001126            CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007534  001401                            BEQ     TST53           ;;

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

007536  104005                          ERROR   5               ;/SIDE 2 TO SIDE 1 DATA
                                                                ;/XFERR ERROR VIA SILO.

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


                                ;;*****************************************************************
                                ;*TEST 53        *TEST THAT DATA PATTERN 100 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE
                                ;;*****************************************************************
007540  000004                  TST53:  SCOPE
                                                                ;/-SIDA-
007542  012737  000100  001124            MOV     #100,$GDDAT     ;/RECORD XFER 100
007550  113777  001124  171632            MOVB    $GDDAT,@MBSEL3 ;/DATA TO XFERR.
007556  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                          .MD.=5
                                                                ;/WRITE TO SILO.
007562  000005                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007564  004537  011656                    JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007570  001424                          .WORD   KMCSR
        000004                          .MD.=4
                                                                ;/READ SILO.
007572  000004                          .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007574  117737  171640  001126            MOVB    @BSEL3,$BDDAT ;/READ FROM REG.
007602  123737  001124  001126            CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007610  001401                            BEQ     TST54           ;;

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

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

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


                                ;;**************************************************************
                                ;*TEST 54        *TEST THAT DATA PATTERN 200 CAN BE XFERRED VIA SILO FROM SIDE 2 TO SIDE
                                ;;**************************************************************
007614  000004                TST54:  SCOPE
                                                                ;/-SIDA-
007616  012737  000200  001124        MOV     #200,$GDDAT     ;/RECORD XFER 200
007624  113777  001124  171556        MOVB    $GDDAT,@MBSEL3  ;/DATA TO XFERR.
007632  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000005                        .MD.=5
                                                                ;/WRITE TO SILO.
007636  000005                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .
                                                                ;/-MCMD-
007640  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007644  001424                        .WORD   KMCSR
        000004                        .MD.=4
                                                                ;/READ SILO.
007646  000004                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND .

007650  117737  171564  001126        MOVB    @BSEL3,$BDDAT   ;/READ FROM REG.
007656  123737  001124  001126        CMPB    $GDDAT,$BDDAT   ;/DATA WRITTEN = DATA RECEIVED?
007664  001401                        BEQ     TST55           ;;

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

007666  104005                        ERROR   5               ;/SIDE 2 TO SIDE 1 DATA
                                                                ;/XFERR ERROR VIA SILO.

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


                                ;;**************************************************************
                                ;*TEST 55        *TEST THAT 7/8 FLAG SETS IN SIDE 1 WHEN 56 WORDS WRITTEN IN SILO
                                ;;**************************************************************
007670  000004                TST55:  SCOPE
007672  012737  000010  001160        MOV     #10,$TIMES      ;;DO 10 ITERATIONS
                                                                ;/-S78F-

007700  012700  000067                MOV     #55.,R0         ;SET TO DO 56 TIMES

007704                          1$:
                                                                ;/-MCMD-
007704  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
007710  001424                        .WORD   KMCSR
```

```
                000001                           .MD.=1
                                                                        ;/COMMAND=NOP.
        007712  000001                           .WORD   .MD.
                                                 ;/RETURN HERE AFTER COMMAND .
        007714  132777  000001  171522           BITB    #BIT0,@BSELS   ;/FIFO OUTPUT READY?
        007722  001370                           BNE     1$
                                                                        ;/-MCMD-
        007724  004537  011656                   JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
        007730  001424                           .WORD   KMCSR
                000005                           .MD.=5
                                                                        ;/WRITE TO SILO.
        007732  000005                           .WORD   .MD.
                                                 ;/RETURN HERE AFTER COMMAND .
        007734  005300                           DEC     R0
        007736  001362                           BNE     1$             ;/DONE 55 TIMES?

        007740  012737  000050  001124           MOV     #50,$GDDAT     ;/EXP'D IOSR
                                                                        ;/-MCMD-
        007746  004537  011656                   JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
        007752  001424                           .WORD   KMCSR
                000001                           .MD.=1
                                                                        ;/COMMAND=NOP.
        007754  000001                           .WORD   .MD.
                                                 ;/RETURN HERE AFTER COMMAND .
        007756  117737  171462  001126           MOVB    @BSELS,$BDDAT  ;/READ SR.
        007764  132777  000200  171452           BITB    #BIT7,@BSELS ;/DID 7/8 FLAG SET?
        007772  001404                           BEQ     2$             ;/NO - GOOD NEED 1 MORE CHARACTER

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

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

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

        007776  004737  012524                   JSR     PC,BINT
        010002  000452                           BR      TST56          ;;

        010004                           2$:
                                                                        ;/-MCMD-
        010004  004537  011656                   JSR     R5,ISSUEC      ;/ISSUE COMMAND TO KMC #1.
        010010  001424                           .WORD   KMCSR
                000005                           .MD.=5
                                                                        ;/WRITE TO SILO.
        010012  000005                           .WORD   .MD.
                                                 ;/RETURN HERE AFTER COMMAND .

        010014  052737  000200  001124           BIS     #BIT7,$GDDAT   ;/EXP'D SR BIT 7 SET.
                                                                        ;/-MCMD-
        010022  004537  011656                   JSR     R5,ISSUEC      ;/ISSUE COMMAND TO kMC #1.
        010026  001424                           .WORD   KMCSR
                000001                           .MD.=1
                                                                        ;/COMMAND=NOP.
        010030  000001                           .WORD   .MD.
```

```
                                        ;/RETURN HERE AFTER COMMAND.
010032  117737  171406  001126        MOVB    @BSEL5,$BDDAT   ;/READ SR.
010040  132777  000200  171376        BITB    #BIT7,@BSEL5 ;/DID 7/8 FLAG SET NOW?
010046  001004                       BNE     3$              ;/IF YES - GOOD.

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

010050  104001                       ERROR   1               ;/7/8 FLAG FAILED TO SET IN SIDE 1
                                        ;/AFTER 56. WORDS WRITTEN

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

010052  004737  012524                JSR     PC,BINT
010056  000424                       BR      TST56           ;;

010060                          3$:
010060  004537  012100                JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000004                        .MD.=4
                                        ;/READ SILO.
010064  000004                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND.
                                        ;/-MCMD-
010066  004537  011656                JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
010072  001424                        .WORD   KMCSR
        000001                        .MD.=1
                                        ;/COMMAND=NOP.
010074  000001                        .WORD   .MD.
                                        ;/RETURN HERE AFTER COMMAND.
010076  117737  171342  001126        MOVB    @BSEL5,$BDDAT   ;/READ CSR.
010104  042737  000200  001124        BIC     #BIT7,$GDDAT            ;/EXPECT BIT 7 CLEAR.
010112  132777  000200  171324        BITB    #BIT7,@BSEL5 ;/7/8 FULL FLAG SHOULD BE CLEAR.
010120  001401                       BEQ     4$              ;/IF CLEAR, GOOD.

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

010122  104001                       ERROR   1               ;/7/8 FLAG ON SIDE 1
                                        ;/FAILED TO CLEAR 56 WORDS WRITTEN
                                        ;/ONE WORD READ; COUNT = 55.

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

010124  004737  012524          4$:   JSR     PC,BINT

                                ;;**********************************************************************
                                ;*TEST 56        *TEST THAT 7/8 FLAG SETS IN SIDE 2 WHEN 56 WORDS WRITTEN IN SILO
                                ;;**********************************************************************
010130  000004                  TST56:  SCOPE
010132  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
                                        ;/-S78F-

010140  012700  000067                MOV     #55.,R0         ;SET TO DO 56 TIMES

010144                          1$:
010144  004537  012100                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,@MBSEL5   ;/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 SILO.
2883  010166  000005                               .WORD   .MD.
2884                                                ;/RETURN HERE AFTER COMMAND .
2885  010170  005300                               DEC     R0
2886  010172  001364                               BNE     1$              ;/DONE 55 TIMES?
2887
2888  010174  012737  000054  001124               MOV     #54,$GDDAT      ;/EXP'D UBSR
2889  010202  004537  012100                       JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2890                000001                          .MD.=1
2891                                                ;/COMMAND=NOP.
2892  010206  000001                               .WORD   .MD.
2893                                                ;/RETURN HERE AFTER COMMAND .
2894  010210  117737  171200  001126               MOVB    @MBSEL5,$BDDAT  ;/READ SR.
2895  010216  132777  000200  171170               BITB    #BIT7,@MBSEL5 ;/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                          2$:
2909  010236  004537  012100                       JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
2910                000005                          .MD.=5
2911                                                ;/WRITE TO SILO.
2912  010242  000005                               .WORD   .MD.
2913                                                ;/RETURN HERE AFTER COMMAND .
2914
2915  010244  052737  000200  001124               BIS     #BIT7,$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  010256  000001                               .WORD   .MD.
2920                                                ;/RETURN HERE AFTER COMMAND .
2921  010260  117737  171130  001126               MOVB    @MBSEL5,$BDDAT  ;/READ SR.
2922  010266  132777  000200  171120               BITB    #BIT7,@MBSEL5 ;/DID 7/8 FLAG SET NOW?
2923  010274  001004                               BNE     3$              ;/IF YES - GOOD.

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

2927  010276  104002                               ERROR   2               ;/7/8 FLAG FAILED TO SET IN SIDE 2
```

```
                                                        ;/AFTER 56. WORDS WRITTEN
                                                ;**********>> ERROR <<**********
010300  004737  012524                          JSR     PC,BINT
010304  000424                                  BR      TST57                   ;;
010306                              3$:
                                                        ;/-MCMD-
010306  004537  011656                          JSR     R5,ISSUEC               ;/ISSUE COMMAND TO KMC #1.
010312  001424                                  .WORD   KMCSR
        000004                                  .MD.=4
                                                        ;/READ SILO.
010314  000004                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND
010316  004537  012100                          JSR     R5,KMSIM                ;/DMC SIMULATE KMC INSTR. LIST FOR.
        000001                                  .MD.=1
                                                        ;/COMMAND=NOP.
010322  000001                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
010324  117737  171064  001126                  MOVB    @MBSEL5,$BDDAT  ;/READ CSR.
010332  042737  000200  001124                  BIC     #BIT7,$GDDAT            ;/EXPECT BIT 7 CLEAR.
010340  132777  000200  171046                  BITB    #BIT7,@MBSEL5 ;/7/8 FULL FLAG SHOULD BE CLEAR.
010346  001401                                  BEQ     4$                      ;/IF CLEAR, GOOD.
                                                ;**********>> ERROR <<**********
010350  104002                                  ERROR   2                       ;/7/8 FLAG ON SIDE 2
                                                                                ;/FAILED TO CLEAR 56 WORDS WRITTEN
                                                                                ;/ONE WORD READ; COUNT = 55.
                                                ;**********>> ERROR <<**********
010352  004737  012524                  4$:     JSR     PC,BINT


                                                ;;**************************************************************
                                                ;*TEST 57        *TEST THAT SIDE 1 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRI
                                                ;;**************************************************************
010356  000004                          TST57:  SCOPE
010360  012737  000010  001160                  MOV     #10,$TIMES              ;;DO 10 ITERATIONS
                                                        ;/-SOVFL-
010366  012700  000100                          MOV     #64.,R0                 ;/SET TO DO 64 XFERRS.
010372                              1$:
                                                        ;/-MCMD-
010372  004537  011656                          JSR     R5,ISSUEC               ;/ISSUE COMMAND TO KMC #1.
010376  001424                                  .WORD   KMCSR
        000001                                  .MD.=1
                                                        ;/COMMAND=NOP.
010400  000001                                  .WORD   .MD.
                                                ;/RETURN HERE AFTER COMMAND .
```

```
2982  010402  132777  000001  171034          BITB    #BIT0,@BSEL5    ;/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,$GDDAT     ;/EXP'D IOSR.
2994
2995                                                                  ;/-MCMD-
2996  010434  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO kMC #1.
2997  010440  001424                          .WORD   KMCSR
2998          000001                          .MD.=1
2999                                                                  ;/COMMAND=NOP.
3000  010442  000001                          .WORD   .MD.
3001                                          ;/RETURN HERE AFTER COMMAND.
3002  010444  132777  000100  170772          BITB    #BIT6,@BSEL5 ;/FIFO R/W ERROR SET?
3003  010452  001407                          BEQ     2$              ;/NO - GOOD
3004  010454  117737  170764  001126          MOVB    @BSEL5,$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$:
3018                                                                  ;/-MCMD-
3019  010472  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
3020  010476  001424                          .WORD   KMCSR
3021          000005                          .MD.=5
3022                                                                  ;/WRITE TO SILO.
3023  010500  000005                          .WORD   .MD.
3024                                          ;/RETURN HERE AFTER COMMAND.
3025                                                                  ;/-MCMD-
3026  010502  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
3027  010506  001424                          .WORD   KMCSR
3028          000001                          .MD.=1
3029                                                                  ;/COMMAND=NOP.
3030  010510  000001                          .WORD   .MD.
3031                                          ;/RETURN HERE AFTER COMMAND.
3032  010512  052737  000100  001124          BIS     #BIT6,$GDDAT    ;/EXPECT BIT 6 TO SET.
3033  010520  117737  170720  001126          MOVB    @BSEL5,$BDDAT   ;/READ SR.
3034  010526  132777  000100  170710          BITB    #BIT6,@BSEL5 ;/FIFO OVFL SET?
3035  010534  001007                          BNE     3$              ;/YES - GOOD.
```

```
3036  010536  117737  170702  001126          MOVB    @BSEL5,$BDDAT   ;/RECORD SR

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

3040  010544  104001                          ERROR   1               ;/FIFO R/W ERROR FLAG FAILED TO SET
3041                                                                  ;/IN SIDE 1 WHEN 65.
3042                                                                  ;/WORDS WRITTEN.

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

3046  010546  004737  012524                  JSR     PC,BINT
3047  010552  000426                          BR      TST60           ;;
3048
3049  010554  004737  012524          3$:     JSR     PC,BINT         ;/NOW INITIALIZE.
3050                                                                  ;/-MCMD-
3051  010560  004537  011656                  JSR     R5,ISSUEC       ;/ISSUE COMMAND TO KMC #1.
3052  010564  001424                          .WORD   KMCSR
3053          000001                          .MD.=1
3054                                                                  ;/COMMAND=NOP.
3055  010566  000001                          .WORD   .MD.
3056                                          ;/RETURN HERE AFTER COMMAND .
3057  010570  042737  000300  001124          BIC     #BIT6!BIT7,$GDDAT       ;/OVLF AND 7/8 SHOULD CLEAR.
3058  010576  052737  000002  001124          BIS     #BIT1,$GDDAT            ;/BIT 1 SHOULD SET.
3059  010604  117737  170634  001126          MOVB    @BSEL5,$BDDAT   ;/READ SR.
3060  010612  123737  001124  001126          CMPB    $GDDAT,$BDDAT                   ;/EVERYTHING OK?
3061  010620  001401                          BEQ     4$              ;YES GET OUT.

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

3065  010622  104001                          ERROR   1               ;/FIFO R/W ERROR FLAG FAILED TO
3066                                                                  ;/CLEAR SIDE 1 WHEN
3067                                                                  ;/INITIALIZED.

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

3071
3072  010624  004737  012524          4$:     JSR     PC,BINT
3073
3074                                          ;;*****************************************************************
3075                                          ;*TEST 60        *TEST THAT SIDE 2 READ/WRITE ERROR FLAG SETS WHEN MORE THAN 64 WORDS WRI
3076                                          ;;*****************************************************************
3077  010630  000004                  TST60:  SCOPE
3078  010632  012737  000010  001160          MOV     #10,$TIMES      ;;DO 10 ITERATIONS
3079
3080                                                                  ;/-SOVFL-
3081
3082  010640  012700  000100                  MOV     #64.,R0         ;/SET TO DO 64 XFERRS.
3083
3084  010644                          1$:
3085  010644  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3086          000001                          .MD.=1
3087                                                                  ;/COMMAND=NOP.
3088  010650  000001                          .WORD   .MD.
3089                                          ;/RETURN HERE AFTER COMMAND .
```

```
3090  010652  132777  000001  170534          BITB    #BIT0,@MBSEL5   ;/FIFO READY?
3091  010660  001371                          BNE     1$
3092  010662  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3093          000005                          .MD.=5
3094                                          ;/WRITE TO SILO.
3095  010666  000005                          .WORD   .MD.
3096                                          ;/RETURN HERE AFTER COMMAND.
3097  010670  005300                          DEC     R0              ;/DONE 64 TIMES?
3098  010672  001364                          BNE     1$              ;/NO - REPEAT
3099  010674  012737  000256  001124          MOV     #256,$GDDAT     ;/EXP'D UBSR.
3100
3101  010702  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3102          000001                          .MD.=1
3103                                          ;/COMMAND=NOP.
3104  010706  000001                          .WORD   .MD.
3105                                          ;/RETURN HERE AFTER COMMAND.
3106  010710  132777  000100  170476          BITB    #BIT6,@MBSEL5 ;/FIFO R/W ERROR SET?
3107  010716  001407                          BEQ     2$              ;/NO - GOOD
3108  010720  117737  170470  001126          MOVB    @MBSEL5,$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$:
3122  010736  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3123          000005                          .MD.=5
3124                                          ;/WRITE TO SILO.
3125  010742  000005                          .WORD   .MD.
3126                                          ;/RETURN HERE AFTER COMMAND.
3127  010744  004537  012100                  JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3128          000001                          .MD.=1
3129                                          ;/COMMAND=NOP.
3130  010750  000001                          .WORD   .MD.
3131                                          ;/RETURN HERE AFTER COMMAND.
3132  010752  052737  000100  001124          BIS     #BIT6,$GDDAT    ;/EXPECT BIT 6 TO SET.
3133  010760  117737  170430  001126          MOVB    @MBSEL5,$BDDAT  ;/READ SR.
3134  010766  132777  000100  170420          BITB    #BIT6,@MBSEL5 ;/FIFO OVFL SET?
3135  010774  001007                          BNE     3$              ;/YES - GOOD.
3136  010776  117737  170412  001126          MOVB    @MBSEL5,$BDDAT  ;/RECORD SR

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

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

```
                                        ;**********>> 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/FRO DATA.
3186  011106  000036                    .WORD   30.             ;/BYTE COUNT FOR XFERR.
3187                                    ;/RETURN HERE AFTER COMMAND .
3188  011110  012700  017722            MOV     #RECBUF,R0
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.
```

```
3198            000004                          .MD.=4
3199                                                            ;/READ SILO.
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$:      MOV     #RECBUF,R0      ;POINT TO RECEIVED DATA
3229   011222   012701   017322                MOV     #SENBUF,R1      ;POINT TO GOOD DATA
3230   011226   012702   000076                MOV     #62.,R2         ;# OF DATA TO CHECK BYTES
3231   011232   122021               2$:      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               3$:      DEC     R2              ;CHECK ALL DATA?
3246   011254   001366                         BNE     2$              ;NO - DO NEXT ONE.
3247
3248                                            ;;*****************************************************************
3249                                            ;*TEST 62        *TEST BURST MODE XFERR SIDE 2 TO SIDE 1
3250                                            ;;*****************************************************************
3251   011256   000004               TST62:   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          000003                            .MD.=3
3257                                                                      ;/READ SILO, NPR WRITE TO MEM.
3258          000603                            .MD.=.MD.!6GO
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$:      MOV     #SENBUF,R0      ;ADDR. OF DATA
3264  011304  012701  000076                    MOV     #62.,R1         ;#OF WORDS TO x FERR
3265  011310  112077  170074          20$:      MOVB    (0)+,@MBSEL3    ;XFER BYTE
3266  011314  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3267          000005                            .MD.=5
3268                                                                      ;/WRITE TO SILO.
3269  011320  000005                            .WORD   .MD.
3270                                            ;/RETURN HERE AFTER COMMAND .
3271  011322  005301                            DEC     R1              ;ALL DONe?
3272  011324  001371                            BNE     20$             ;NO-CONTINUE
3273  011326  004537  012100                    JSR     R5,KMSIM        ;/DMC SIMULATE KMC INSTR. LIST FOR.
3274          000001                            .MD.=1
3275                                                                      ;/COMMAND=NOP.
3276  011332  000001                            .WORD   .MD.
3277                                            ;/RETURN HERE AFTER COMMAND .
3278  011334  005037  001126                    CLR     $BDDAT
3279  011340  117737  170100  001126            MOVB    @BSEL5,$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$:       MOV     #RECBUF,R0      ;POINT TO RECEIVED DATA
3297  011400  012701  017322                    MOV     #SENBUF,R1      ;POINT TO GOOD DATA
3298  011404  012702  000076                    MOV     #62.,R2         ;# OF DATA TO CHECK BYTES
3299  011410  122021                  2$:       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 <<**********
```

```
011424  104005                          ERROR   5               ;BUST XFERR THROUGH SILO
                                                                ;SIDE 2 TO SIDE 1 PRODUCED
                                                                ;A DATA ERROR.

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

011426  000402                          BR      TST63           ;;
011430  005302                  3$:     DEC     R2              ;CHECK ALL DATA?
011432  001366                          BNE     2$              ;NO - DO NEXT ONE.

                                ;;**************************************************************
                                ;*TEST 63            END OF TESTS
                                ;;**************************************************************
011434  000004                  TST63:  SCOPE

011436  023737  001372  001366          CMP     NCNT,MCNT       ;DONE ALL?
011444  001410                          BEQ     10$
011446  005237  001372                  INC     NCNT            ;DO NEXT.
011452  013700  001424                  MOV     KMCSR,R0
011456  063700  001370                  ADD     VAR2,R0
011462  000137  001724                  JMP     LSTED
011466                          10$:

                                .SBTTL  END OF PASS ROUTINE

                                ;;**************************************************************
                                ;*INCREMENT THE PASS NUMBER ($PASS)
                                ;*IF THERES A MONITOR GO TO IT
                                ;*IF THERE ISN'T JUMP TO RTNAD

011466                          $EOP:
011466  000240                          NOP
011470  005037  001102                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
011474  005037  001160                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
011500  005237  001200                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
011504  042737  100000  001200          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
011512  005327                          DEC     (PC)+           ;;LOOP?
011514  000001                  $EOPCT: .WORD   1
011516  003053                          BGT     $DOAGN          ;;YES
011520  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
011522  000001                  $ENDCT: .WORD   1
011524  011514                          $EOPCT

                                                                ;/-ENDPAS-
011526  104401  011534                  TYPE    ,65$            ;;TYPE ASCIZ STRING
011532  000406                          BR      64$             ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200>#END PASS #
011550                          64$:
011550  013746  001200                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
                                                                ;;TYPE PASS NUMBER.
011554  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011556  104401  011564                  TYPE    ,67$            ;;TYPE ASCIZ STRING
011562  000411                          BR      66$             ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  # ; TOTAL ERRORS #
```

```
 3360  011606                          66$:
 3361  011606  013746  001456                MOV     ERCNT,-(SP)     ;;SAVE ERCNT FOR TYPEOUT
 3362  011612  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
 3363
 3364  011614  104401  011622                TYPE    ,69$            ;;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     377  000  $ENULL: .BYTE   -1,-1,0          ;;NULL CHARACTER STRING
 3380           011656                       .EVEN
 3381                                  ;*
 3382                                  ;*COMMAND ROUTINE
 3383                                  ;*THIS ROUTINE WILL ISSUE COMMANDS TO THE
 3384                                  ;*SPECIFIED KMC-11 AND WAITE FOR COMMAND COMPLETION
 3385                                  ;*UNLESS INHIBITED BY COMMAND CALL.
 3386                                  ;*
 3387                                  ;*CALL= JSR     R5,ISSUEC
 3388                                  ;*      ARG1
 3389                                  ;*      ARG2
 3390                                  ;*      ARG3    (IF NEEDED)
 3391                                  ;*      ARG4    (IF NEEDED)
 3392                                  ;*      -> RETURNS HERE <-
 3393                                  ;*ARG1  SPECIFIES THE ADDRESS OF THE ADDRESS OF THE KMC-11
 3394                                  ;*ARG2  SPECIFIES THE COMMAND FOR THE KMC-11 FIREMWARE
 3395                                  ;*      IN BITS 0-6
 3396                                  ;*      IN BIT7 WE SPECIFY IF ARG3 AND 4 ARE USED
 3397                                  ;*      IN BIT8 WE SPECIFY IF WE SHOULD WAIT FOR
 3398                                  ;*      COMMAND COMPLETION BEFORE RETURNING
 3399                                  ;*ARG3  SPECIFIES BSEL2+BSEL3 INFO (ADDRESS).
 3400                                  ;*ARG4  SPECIFIES BSEL4 INFO. (BYTE COUNT).
 3401                                  ;*
 3402
 3403  011656  010046                ISSUEC: MOV     R0,-(SP)        ;SAVE R0
 3404  011660  013500                        MOV     @(5)+,R0        ;PICK UP KMC ADDR.
 3405  011662  105715                        TSTB    (5)             ;EXTENDED COMMAND?
 3406  011664  100030                        BPL     1$              ;NO-AHEAD.
 3407  011666  116560  000002  000002        MOVB    2(5),2(0)       ;SET BSEL2
 3408  011674  116560  000003  000003        MOVB    3(5),3(0)       ;SET BSEL3
 3409  011702  116560  000004  000004        MOVB    4(5),4(0)       ;SET BSEL4
 3410  011710  142715  000200                BICB    #200,(5)
 3411  011714  111560  000000                MOVB    (5),0(0)        ;SET COMMAND
 3412  011720  152715  000200                BISB    #200,(5)
 3413  011724  062705  000006                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          011770          RBST=.
3428
3429  011770  104401  011776                  TYPE    ,65$            ;;TYPE ASCIZ STRING
3430  011774  000415                          BR      64$             ;;GET OVER THE ASCIZ
3431                          ;;65$:  .ASCIZ  <200>#MICRO-HUNG--RESTARTING.#
3432  012030                  64$:
3433  012030  012706  001100                  MOV     #STACK,SP       ;RESET STACK.
3434  012034  004737  012524                  JSR     PC,BINT         ;INITIALIZE MICRO PROCESSORS.
3435  012040  000177  167042                  JMP     @$LPADR         ;RESTART SUB TEST.
3436
3437  012044  012600          ISSEX:  MOV     (SP)+,R0        ;RESTORE R0
3438  012046  000205                  RTS     R5              ;EXIT
3439
3440  012050  122710  000377          ERRIS:  CMPB    #377,(0)        ;IF CMD=377 THE KMC HAS FINISHED
3441  012054  001773                          BEQ     ISSEX           ;THE COMMAND
3442  012056  010037  001462                  MOV     R0,KLAD         ;STORE KMC ADDR.
3443  012062  116037  000005  001126          MOVB    5(0),$BDDAT     ;GET CSR
3444  012070  105037  001127                  CLRB    $BDDAT+1        ;UPPER BYTE UNUSED

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

3448  012074  104011                  ERROR   11              ;MICRO PROCESSOR PASSED
3449                                                          ;BACK A DETECTED ERROR
3450                                                          ;SEE TEST WE WERE
3451                                                          ;DOInG AT TIME.

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

3455
3456  012076  000762                  BR      ISSEX
3457
3458
3459
3460                                    ;KMSIM  ←THIS ROUTINE IS USED TO MAKE THE M8200-YC ACT LIKE A
3461                                    ;        KMC-11 THE M8200-YC, EVEN THOUGH (P)ROM PROGRAM BLASTED,
3462                                    ;        MAY EXECUTE SINGLE INSTRUCTIONS, ONE AT A TIME. WHEN WE
3463                                    ;        WILL PASS A NUMBER THAT REPRESENTS THE SEQUENCE OF
3464                                    ;        INTRUCTIONS WE WISH IT TO DO. WE THEN TAKE THIS NUMBER
3465                                    ;        AND USE IT AS A GUIDE TO START EXECUTING THE CODE
3466                                    ;        LOADING INTO THE KMC-11.
3467                                    ;
```

```
                                                ;       CALL=   JSR     R5,KMSIM
                                                ;               .WORD   X       (COMMAND TO PERFORM)
                                                ;               ;RETURNS HERE WHEN DONe
                                        KMSIM:  MOV     R0,-(SP)        ;SAVE R0
                                                MOV     R1,-(SP)        ;SAVE R1
                                                MOV     (5)+,   R0      ;PICK-UP COMMAND #.
012100  010046  177700                          BIC     #177700,R0      ;STRIP ANY JUNK.
012102  010146                                  ASL     R0              ;ADJUST POINTER.
012104  012500                                  ADD     #MRCODE,R0      ;ADDR. OF COMMAND LIST
012106  042700  000000G                         CLR     R1
012112  006300                                  MOVB    (0),R1          ;POINTER TO INSTRUCTION LIST.
012114  062700  000000G                         ASL     R1              ;POINTER=POINTER TIMES 2
012120  005001                                  BIC     #177000,R1
012122  111001                                  ADD     #MRCODE,R1      ;PLUS LIST
012124  006301                                                          ;POINTER TO LIST OF INSTRUCTIONS
012126  042701  177000                          ;IS R1<-
012132  062701  000000G                 1$:     MOV     (1)+,R0         ;GET INSTUCTION.
                                                BMI     2$              ;IF A BRANCH INSTRUCTION,EXIT
012136  012100                          3$:     JSR     PC,DOIT         ;IF NOT-MAKE DMC-EXECUTE IT.
012140  100403
012142  004737  012166
                                                BR      1$
012146  000773                          2$:     BIT     #070000,-2(1)   ;MAKE SURE IT WAS A BR INSTR.
012150  032761  070000  177776                  BNE     3$              ;IF A MOVE,GO DO IT.
012156  001371
                                                MOV     (SP)+,R1        ;IT WAS A BR RESTORE R1
012160  012601                                  MOV     (SP)+,R0        ;AND R0.
012162  012600                                  RTS     R5              ;THEN EXIT.
012164  000205
                                        DOIT:   MOVB    #BIT01,@MBSEL1          ;SET ROMI
012166  112777  000002  167210                  NOP
012174  000240                                  MOV     R0,@MBSEL6              ;LOAD INSTR.
012176  010077  167214                          NOP
012202  000240                                  BISB    #BIT1!BIT0,@MBSEL1      ;CLOCK INSTRUCTION
012204  152777  000003  167172                  NOP
012212  000240                                  NOP
012214  000240                                  NOP
012216  000240                                  NOP
012220  000240                                  NOP
012222  000240                                  NOP
012224  000240                                  BICB    #BIT2!BIT1!BIT0,@MBSEL1 ;CLEAR ROM0,ROMI,SET
012226  142777  000007  167150                  RTS     PC
012234  000207
                                                ;TAKES CARE OF CLOCK INTERRUPTS.
                                        KWINT:  INC     TIME            ;UPDATE TIME.
012236  005237  012244                          RTI                     ;EXIT
012242  000002                          TIME:   .WORD   0               ;SET TO MINUS NUMBER BY USER.
012244  000000                                  ;*
                                                ;*LOAD  THIS ROUTINE LOADS THE UCODE
                                                ;*      INTO THE KMC-11 UPROCESSOR.
                                                ;*      CALL= JSR R5,LOAD
                                                ;*
                                                ;*MAKE SURE JUMPER W1 IS REMOVED ON 8254 MODULE!!!
                                                ;*
```

```
 3522  012246  005004                  LOAD:   CLR     R4              ;CLEAR ERROR TIMER
 3523  012250  012700  000000G          3$:     MOV     #MRCODE,R0      ;GET ADDRESS OF UCODE
 3524                                                                  ;CLEAR CSRS.
```

```
3525  012254  005077  167144                    CLR     @KMCSR
3526                                                      ;SET ADDR 0
3527  012260  005077  167142                    CLR     @KMADR          ;IN BOTH KMC-11S
3528
3529  012264  052777  002000  167132  1$:       BIS     #2000,@KMCSR    ;SELECT CRAM.
3530
3531  012272  012077  167132                    MOV     (0)+,@KMDBR     ;WRITE DATA
3532
3533  012276  052777  020000  167120            BIS     #20000,@KMCSR   ;SET CRAM WRITE
3534                                                      ;NOTE:  IF PROGRAM SEEMS TO STOP IN THIS AREA,
3535                                                      ;       MAKE SURE JUMPER W1 ON 8254
3536                                                      ;       MODULE IS REMOVED!!!
3537
3538  012304  005077  167114                    CLR     @KMCSR          ;DISABLE CRAM
3539  012310  005277  167112                    INC     @KMADR          ;IN UNIT
3540  012314  020027  000000G                   CMP     R0,#UCODEE      ;DONE ALL UCODE?
3541  012320  001361                            BNE     1$              ;NO - DO NEXT INSTR.
3542
3543                                                      ;NOW LETS CHECK TO MAKE SURE
3544  012322  005077  167100                    CLR     @KMADR          ;UCODE GOT LOADED CORRECTLY.
3545  012326  012700  000000G                   MOV     #MRCODE,R0
3546
3547  012332  052777  002000  167064  2$:       BIS     #2000,@KMCSR    ;SELECT CRAM.
3548  012340  013737  001426  001462            MOV     KMADR,KLAD
3549  012346  022077  167056                    CMP     (0)+,@KMDBR     ;DATA OK?
3550  012352  001013                            BNE     LOADER
3551  012354  005077  167044                    CLR     @KMCSR          ;CLEAR THE CSR
3552  012360  005277  167042                    INC     @KMADR          ;CHANGE UCODE ADDR.
3553  012364  020027  000000G                   CMP     R0,#UCODEE      ;AT END?
3554  012370  001360                            BNE     2$
3555  012372  012777  100000  167024            MOV     #BIT15,@KMCSR   ;SET RUN.
3556  012400  000205                            RTS     R5              ;RETURN.
3557
3558                                            ;COME HERE ON LOAD ERROR
3559
3560          012402            LOADER=.
3561  012402  105204                            INCB    R4              ;MAKE SURE WE DON'T TRY TOO MANY TIMES.
3562  012404  100321                            BPL     3$              ;LOOP BACK TRY AGAIN.
3563
3564  012406  104401  012414                    TYPE    ,65$            ;;TYPE ASCIZ STRING
3565  012412  000415                            BR      64$             ;;GET OVER THE ASCIZ
3566                                    ;;65$:  .ASCIZ  <200>!ERROR IN LOADING UCODE !
3567  012446                            64$:
3568  012446  104401  012454                    TYPE    ,67$            ;;TYPE ASCIZ STRING
3569  012452  000404                            BR      66$             ;;GET OVER THE ASCIZ
3570                                    ;;67$:  .ASCIZ  <200>!GOOD= !
3571  012464                            66$:
3572  012464  005740                            TST     -(0)
3573  012466                            10$:
3574  012466  011046                            MOV     (R0),-(SP)      ;;SAVE (R0) FOR TYPEOUT
3575  012470  104402                            TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3576  012472  104401  012500                    TYPE    ,69$            ;;TYPE ASCIZ STRING
3577  012476  000404                            BR      68$             ;;GET OVER THE ASCIZ
3578                                    ;;69$:  .ASCIZ  !  BAD= !
```

```
3579  012510                          68$:
3580  012510  017746  166714                MOV     @KMDBR,-(SP)    ;;SAVE @KMDBR FOR TYPEOUT
3581  012514  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
3582  012516  000000                        HALT                    ;ERROR IN LOADING UCODE,RETRYS
3583                                                                ;HAVE FAILED. RUN KMC11 DIAGNOSTIC.
3584  012520  000137  012250              JMP     3$
3585
3586                                        ;*
3587                                        ;*THIS ROUTINE WILL ISSUE INITIALIZE TO
3588                                        ;*BOTH KMC11S.
3589                                        ;*CALL= JSR     PC,BINT
3590                                        ;*
3591  012524  000005                BINT:   RESET
3592  012526  012777  100000  166670        MOV     #BIT15,@KMCSR   ;START #1.
3593  012534  005077  166634                CLR     @KMMCSR         ;HALT #2.
3594  012540  000207                        RTS PC
3595                                        ;
3596                                        ;ROUTINE TO FILL SENBUF WITH RANDOM
3597                                        ;DATA
3598                                        ;
3599  012542  012700  017322        FILBUF: MOV     #SENBUF,R0      ;GET ADDRESS
3600
3601
3602  012546  004737  012600        1$:     JSR     PC,RAND         ;GET 2 RANDOM NUMBERS
3603  012552  013720  012626                MOV     RANA,(0)+
3604  012556  020027  017722                CMP     R0,#RECBUF      ;BUFFER FULL?
3605  012562  001405                        BEQ     FILEX
3606  012564  013720  012630                MOV     RANB,(0)+
3607  012570  020027  017722                CMP     R0,#RECBUF      ;BUFFER FULL?
3608  012574  001364                        BNE     1$
3609  012576  000207                FILEX:  RTS     PC
3610
3611  012600  063737  012626  012630  RAND:   ADD     RANA,RANB
3612  012606  005537  012626                ADC     RANA
3613  012612  063737  012630  012626        ADD     RANB,RANA
3614  012620  005537  012630                ADC     RANB
3615  012624  000207                        RTS     PC
3616
3617  012626  123456                RANA:   123456
3618  012630  071234                RANB:   071234
3619
3620                                        ;ROUTINE TO READ/MODIFY CRAM.
3621                                        ;L&S AT 210
3622
3623  012632                        MOCRAM:
3624  012632  104401  012640                TYPE    ,65$            ;;TYPE ASCIZ STRING
3625  012636  000406                        BR      64$             ;;GET OVER THE ASCIZ
3626                                ;;65$:  .ASCIZ  <200>#KMC ADR? #
3627  012654                        64$:
3628  012654  104412                        RDOCT
3629  012656  012600                        MOV     (SP)+,R0
3630  012660  052700  160000                BIS     #160000,R0
3631  012664                        1$:
3632  012664  104401  012672                TYPE    ,67$            ;;TYPE ASCIZ STRING
```

```
3633  012670  000407                        BR      66$                 ;;GET OVER THE ASCIZ
3634                                ;;67$:    .ASCIZ  <200>#CRAM ADDR? #
3635  012710                        66$:
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)
```

```
012744  010160  000004                  MOV     R1,4(0)                 ;SET ADDR.
012750  052710  002000                  BIS     #2000,(0)               ;SELECT CRAM.
012754  016002  000006                  MOV     6(0),R2                 ;READ CRAM.
012760  010246                          MOV     R2,-(SP)                ;;SAVE R2 FOR TYPEOUT
012762  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
012764  104401  012772                  TYPE    ,73$                    ;;TYPE ASCIZ STRING
012770  000403                          BR      72$                     ;;GET OVER THE ASCIZ
                                ;;73$:  .ASCIZ  # \C\ #
013000                          72$:
013000  105777  166140          10$:    TSTB    @$TKS
013004  100375                          BPL     10$
013006  117703  166134                  MOVB    @$TKB,R3
013012  042703  000200                  BIC     #200,R3
013016  120327  000012                  CMPB    R3,#12
013022  001002                          BNE     11$
013024  005201                          INC     R1
013026  000732                          BR      9$
013030  120327  000015          11$:    CMPB    R3,#15
013034  001713                          BEQ     1$

013036  104401  013044                  TYPE    ,75$                    ;;TYPE ASCIZ STRING
013042  000402                          BR      74$                     ;;GET OVER THE ASCIZ
                                ;;75$:  .ASCIZ  # / #
013050                          74$:
013050  104412                          RDOCT
013052  012602                          MOV     (SP)+,R2                ;NEW DATA
013054  001703                          BEQ     1$                      ;=0,NO NEW DATA
013056  010260  000006                  MOV     R2,6(0)
013062  052710  020000                  BIS     #20000,(0)              ;CRAM WRITE.
013066  000676                          BR      1$

                                        ;ODT FOR USE ON 11/34
                                        ;CALL= L&S 220
013070                          ODT:
013070  104401  013076                  TYPE    ,65$                    ;;TYPE ASCIZ STRING
013074  000404                          BR      64$                     ;;GET OVER THE ASCIZ
                                ;;65$:  .ASCIZ  <200>#ADDR? #
013106                          64$:
013106  104412                          RDOCT

013110  012600                          MOV     (SP)+,R0
013112                          1$:
013112  104401  013120                  TYPE    ,67$                    ;;TYPE ASCIZ STRING
013116  000401                          BR      66$                     ;;GET OVER THE ASCIZ
                                ;;67$:  .ASCIZ  <200>##
013122                          66$:
013122  010046                          MOV     R0,-(SP)                ;;SAVE R0 FOR TYPEOUT
013124  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013126  104401  013134                  TYPE    ,69$                    ;;TYPE ASCIZ STRING
013132  000402                          BR      68$                     ;;GET OVER THE ASCIZ
                                ;;69$:  .ASCIZ  #/ #
013140                          68$:
013140  011046                          MOV     (0),-(SP)               ;;SAVE (0) FOR TYPEOUT
013142  104402                          TYPOC                           ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
```

```
3704  013144
3705  013144  104401  013152          5$:     TYPE    ,71$            ;;TYPE ASCIZ STRING
3706  013150  000402                          BR      70$             ;;GET OVER THE ASCIZ
3707                                  ;;71$:  .ASCIZ  #\?\#
3708  013156                          70$:
3709  013156  105777  165762          2$:     TSTB    @$TKS
3710  013162  100375                          BPL     2$
3711  013164  117701  165756                  MOVB    @$TKB,R1
3712  013170  042701  000200                  BIC     #200,R1
3713  013174  122701  000015                  CMPB    #15,R1
3714  013200  001733                          BEQ     ODT
3715  013202  122701  000012                  CMPB    #12,R1
3716  013206  001003                          BNE     3$
3717  013210  062700  000002                  ADD     #2,R0
3718  013214  000736                          BR      1$
3719  013216  122701  000033          3$:     CMPB    #33,R1
3720  013222  001003                          BNE     4$
3721  013224  162700  000002                  SUB     #2,R0
3722  013230  000730                          BR      1$
3723  013232                          4$:
3724  013232  104401  013240                  TYPE    ,73$            ;;TYPE ASCIZ STRING
3725  013236  000403                          BR      72$             ;;GET OVER THE ASCIZ
3726                                  ;;73$:  .ASCIZ  #NEW=#
3727  013246                          72$:
3728  013246  104412                          RDOCT
3729  013250  012610                          MOV     (SP)+,(0)
3730  013252  000734                          BR      5$
3731
3732                                  .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
3733
3734                                  ;;**************************************************************
3735                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
3736                                  ;*OCTAL (ASCII) NUMBER AND TYPE IT.
3737                                  ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
3738                                  ;*CALL:
3739                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3740                                  ;*      TYPOS                   ;;CALL FOR TYPEOUT
3741                                  ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
3742                                  ;*      .BYTE   M               ;;M=1 OR 0
3743                                  ;*                                      ;;1=TYPE LEADING ZEROS
3744                                  ;*                                      ;;0=SUPPRESS LEADING ZEROS
3745                                  ;*
3746                                  ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
3747                                  ;*$TYPOS OR $TYPOC
3748                                  ;*CALL:
3749                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3750                                  ;*      TYPON                   ;;CALL FOR TYPEOUT
3751                                  ;*
3752                                  ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
3753                                  ;*CALL:
3754                                  ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
3755                                  ;*      TYPOC                   ;;CALL FOR TYPEOUT
3756
3757  013254  017646  000000          $TYPOS: MOV     @(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  $TYPOC: MOVB    #1,$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$              ;;GO DO MSB
3777  013364  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
3778  013366  006105                          ROL     R5
3779  013370  006105                          ROL     R5
3780  013372  010503                          MOV     R5,R3
3781  013374  006103                  3$:     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$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
3789  013420  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
3790  013424  052703  000040          5$:     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$:     MOV     (SP)+,R5        ;;RESTORE R5
3799  013456  012604                          MOV     (SP)+,R4        ;;RESTORE R4
3800  013460  012603                          MOV     (SP)+,R3        ;;RESTORE R3
3801  013462  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
3802  013470  012616                          MOV     (SP)+,(SP)
3803  013472  000002                          RTI                     ;;RETURN
3804  013474     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
3805  013475     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
3806  013476     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
3807  013477     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
3808  013500  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO tYPE
3809                          .SBTTL  CONVERT BINARY TO DECiMAL AND TYPE ROUTINE
3810
3811                          ;;**************************************************************
```

```
                                        ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
                                        ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
                                        ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
                                        ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
                                        ;*REPLACED WITH SPACES.
                                        ;*CALL:
                                        ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
                                        ;*      TYPDS                   ;;GO TO THE ROUTINE

013502                                  $TYPDS:
013502  010046                                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
013504  010146                                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
013506  010246                                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
013510  010346                                  MOV     R3,-(SP)        ;;PUSH R3 ON STACK
013512  010546                                  MOV     R5,-(SP)        ;;PUSH R5 ON STACK
013514  012746  020200                         MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
013520  016605  000020                         MOV     20(SP),R5       ;;GET THE INPUT NUMBER
013524  100004                                BPL     1$              ;;BR IF INPUT IS POS.
013526  005405                                NEG     R5              ;;MAKE THE BINARY NUMBER POS.
013530  112766  000055  000001                MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
013536  005000                          1$:   CLR     R0              ;;ZERO THE CONSTANTS INDEX
013540  012703  013716                         MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
013544  112723  000040                         MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
013550  005002                          2$:   CLR     R2              ;;CLEAR THE BCD NUMBER
013552  016001  013706                         MOV     $DTB1(R0),R1    ;;GET THE CONSTANT
013556  160105                          3$:   SUB     R1,R5           ;;FORM THIS BCD DIGIT
013560  002402                                BLT     4$              ;;BR IF DONE
013562  005202                                INC     R2              ;;INCREASE THE BCD DIGIT BY 1
013564  000774                                BR      3$
013566  060105                          4$:   ADD     R1,R5           ;;ADD BACK THE CONSTANT
013570  005702                                TST     R2              ;;CHECK IF BCD DIGIT=0
013572  001002                                BNE     5$              ;;FALL THROUGH IF 0
013574  105716                                TSTB    (SP)            ;;STILL DOING LEADING 0'S?
013576  100407                                BMI     7$              ;;BR IF YES
013600  106316                          5$:   ASLB    (SP)            ;;MSD?
013602  103003                                BCC     6$              ;;BR IF NO
013604  116663  000001  177777                MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
013612  052702  000060                  6$:   BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
013616  052702  000040                  7$:   BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
013622  110223                                MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
013624  005720                                TST     (R0)+           ;;JUST INCREMENTING
013626  020027  000010                         CMP     R0,#10          ;;CHECK THE TABLE INDEX
013632  002746                                BLT     2$              ;;GO DO THE NEXT DIGIT
013634  003002                                BGT     8$              ;;GO TO EXIT
013636  010502                                MOV     R5,R2           ;;GET THE LSD
013640  000764                                BR      6$              ;;GO CHANGE TO ASCII
013642  105726                          8$:   TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
013644  100003                                BPL     9$              ;;BR IF NO
013646  116663  177777  177776                MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
013654  105013                          9$:   CLRB    (R3)            ;;SET THE TERmINATOR
013656  012605                                MOV     (SP)+,R5        ;;POP STACK INTO R5
013660  012603                                MOV     (SP)+,R3        ;;POP STACK INTO R3
013662  012602                                MOV     (SP)+,R2        ;;POP STACK INTO R2
013664  012601                                MOV     (SP)+,R1        ;;POP STACK INTO R1
```

```
3866  013666  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
3867  013670  104401  013716                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
3868  013674  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
3869  013702  012616                          MOV     (SP)+,(SP)
3870  013704  000002                          RTI                     ;;RETURN TO USER
3871  013706  023420                  $DTBL:  10000.
3872  013710  001750                          1000.
3873  013712  000144                          100.
3874  013714  000012                          10.
3875  013716  000004                  $DBLK:  .BLKW   4
3876                                  .SBTTL  ERROR HANDLER ROUTINE
3877
3878                                  ;;*********************************************************************
3879                                  ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
3880                                  ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
3881                                  ;*AND GO TO $ERrTYP ON ERROR
3882                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
3883                                  ;*SW15=1         HALT ON ERROR
3884                                  ;*SW13=1         INHIBIT ERROR TYPEOUTS
3885                                  ;*SW10=1         BELL ON ERROR
3886                                  ;*CALL
3887                                  ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
3888
3889  013726                          $ERROR:
3890  013726  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
3891  013730  105237  001103          7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
3892  013734  001775                          BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
3893  013736  013777  001102  165176          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
3894  013744  032777  002000  165166          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
3895  013752  001402                          BEQ     1$              ;;NO - SKIP
3896  013754  104401  001162                  TYPE    ,$BELL          ;;RING BELL
3897  013760  005237  001112          1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
3898  013764  011637  001116                  MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
3899  013770  162737  000002  001116          SUB     #2,$ERRPC
3900  013776  117737  165114  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
3901  014004  032777  020000  165126          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
3902  014012  001004                          BNE     20$             ;;SKIP TYPEOUTS
3903  014014  004737  014074                  JSR     PC,$ERRTYP      ;;GO TO USER ERROR ROUTINE
3904  014020  104401  001167                  TYPE    ,$CRLF
3905  014024                          20$:
3906  014024  122737  000001  001212          CMPB    #APTENV,$ENv    ;;RUNNING IN APT MODE
3907  014032  001007                          BNE     2$              ;;NO,SKIP APT ERROR REPORT
3908  014034  113737  001114  014046          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
3909  014042  004737  015620                  JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
3910  014046     000                  21$:    .BYTE   0
3911  014047     000                          .BYTE   0
3912  014050  000777                  22$:    BR      22$             ;;APT ERROR LOOP
3913  014052  005777  165062          2$:     TST     @SWR            ;;HALT ON ERROR
3914  014056  100002                          BPL     3$              ;;SKIP IF CONTINUE
3915  014060  000000                          HALT                    ;;HALT ON ERROR!
3916  014062  104407                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
3917  014064                          3$:
3918  014064  005237  001456          10$:    INC     ERCNT   ;/UPDATE ERROR COUNT.
3919  014070  001775                          BEQ     10$     ;/DON'T LET IT HIT ZERO.
```

```
                                        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.

014074                          $ERRTYP:
014074  104401  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014100  010046                          MOV     R0,-(SP)        ;;SAVE R0
014102  005000                          CLR     R0              ;;PICKUP THE ITEM INDEX
014104  153700  001114                  BISB    @#$ITEMB,R0
014110  001004                          BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
014112  013746  001116                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
014116  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014120  000426                          BR      6$              ;;GET OUT
014122  005300                  1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
014124  006300                          ASL     R0              ;;      WORK FOR THE ERROR TABLE
014126  006300                          ASL     R0
014130  006300                          ASL     R0
014132  062700  001254                  ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
014136  012037  014146                  MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
014142  001404                          BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
014144  104401                          TYPE                    ;;TYPE THE "ERROR MESSAGE"
014146  000000                  2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
014150  104401  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014154  012037  014164          3$:     MOV     @(R0)+,4$       ;;PICKUP "DATA HEADER" POINTER
014160  001404                          BEQ     5$              ;;SKIP TYPEOUT IF 0
014162  104401                          TYPE                    ;;TYPE THE "DATA HEADER"
014164  000000                  4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
014166  104401  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014172  011000                  5$:     MOV     @(R0),R0        ;;PICKUP "DATA TABLE" POINTER
014174  001004                          BNE     7$              ;;GO TYPE THE DATA
014176  012600                  6$:     MOV     (SP)+,R0        ;;RESTORE R0
014200  104401  001167                  TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
014204  000207                          RTS     PC              ;;RETURN
014206                          7$:
014206  013046                          MOV     @(R0)+,-(SP)    ;;SAVE @(R0)+ FOR TYPEOUT
014210  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014212  005710                          TST     (R0)            ;;IS THERE ANOTHER NUMBER?
014214  001770                          BEQ     6$              ;;BR IF NO
014216  104401  014224                  TYPE    ,8$             ;;TYPE TWO(2) SPACES
014222  000771                          BR      7$              ;;LOOP
014224  020040    000           8$:     .ASCIZ  /  /            ;;TWO(2) SPACES
014230                                  .EVEN
                                .SBTTL  SCOPE HANDLER ROUTINE

                                ;;**********************************************************************
                                ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
                                ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
                                ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
```

```
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW14=1          LOOP ON TEST
                                        ;*SW11=1          INHIBIT ITERATIONS
                                        ;*SW08=1          LOOP ON TEST IN SWR<7:0>
                                        ;*CALL
                                        ;*      SCOPE               ;;SCOPE=IOT

014230                                  $SCOPE:
014230  104407                                 CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
014232  104407                                 CKSWR
014234  032777  040000  164676  1$:      BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
014242  001070                          BNE     $OVER           ;;YES IF SW14=1
                                        ;#####START OF CODE FOR THE XOR TESTER#####
014244  000416                          $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
                                                                        ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
014246  013746  000004                          MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
014252  012737  014272  000004                  MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
014260  005737  177060                          TST     @#177060        ;;TIME OUT ON XOR?
014264  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014270  000446                          BR      $SVLAD          ;;GO TO THE NEXT TEST
014272  022626                          5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
014274  012637  000004                          MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
014300  000451                          BR      $OVER           ;;LOOP ON THE PRESENT TEST
014302                                  6$:;#####END OF CODE FOR THE XOR TESTER#####
014302  032777  000400  164630                  BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
014310  001404                          BEQ     2$              ;;BR IF NO
014312  127737  164622  001102                  CMPB    @SWR,$TSTNM     ;;ON THE RIGHT TEST?    SWR<7:0>
014320  001441                          BEQ     $OVER           ;;BR IF YES
014322  105737  001103                  2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
014326  001404                          BEQ     3$              ;;BR IF NO
014330  105037  001103                  4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
014334  005037  001160                          CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
014340  032777  004000  164572  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
014346  001011                          BNE     1$              ;;BR IF YES
014350  005737  001200                          TST     $PASS           ;;IF FIRST PASS OF PROGRAM
014354  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
014356  005237  001104                          INC     $ICNT           ;;INCREMENT ITERATION COUNT
014362  023737  001160  001104                  CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
014370  002015                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
014372  012737  000001  001104  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
014400  013737  014440  001160          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
014406  105237  001102          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
014412  113737  001102  001176          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
014420  011637  001106                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
014424  013777  001102  164510  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
014432  013716  001106                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
014436  000002                          RTI                     ;;FIXES PS
014440  001000                  $MXCNT: 512.                    ;;MAX. NUMBER OF ITERATIONS
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;******************************************************************

                                        .ENABL  LSB

                                        ;;******************************************************************
```

```
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
014442  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
014450  001074                         BNE     15$             ;;BRANCH IF NO
014452  105777  164466                 TSTB    @$TKS           ;;CHAR THERE?
014456  100071                         BPL     15$             ;;IF NO, DON'T WAIT AROUND
014460  117746  164462                 MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
014464  042716  177600                 BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
014470  022726  000007                 CMP     #7,(SP)+        ;;IS IT A CONTROL G?
014474  001062                         BNE     15$             ;;NO, RETURN TO USER
014476  123727  001134  000001         CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
014504  001456                         BEQ     15$             ;;BRANCH IF YES

014506  104401  015167                 TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
014512  104401  015174         $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
014516  013746  000176                 MOV     $SWREG,-(SP)    ;;SAVE SWREG FOR TYPEOUT
014522  104402                         TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
014524  104401  015205                 TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
014530  005046                 19$:    CLR     -(SP)           ;;CLEAR COUNTER
014532  005046                         CLR     -(SP)           ;;THE NEW SWR
014534  105777  164404         7$:     TSTB    @$TKS           ;;CHAR THERE?
014540  100375                         BPL     7$              ;;IF NOT TRY AGAIN

014542  117746  164400                 MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
014546  042716  177600                 BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII


014552  021627  000025         9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
014556  001005                         BNE     10$             ;;BRANCH IF NOT
014560  104401  015162                 TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
014564  062706  000006         20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
014570  000757                         BR      19$             ;;LET'S TRY IT AGAIN


014572  021627  000015         10$:    CMP     (SP),#15        ;;IS IT A <CR>?
014576  001022                         BNE     16$             ;;BRANCH IF NO
014600  005766  000004                 TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
014604  001403                         BEQ     11$             ;;BRANCH IF YES
014606  016677  000002  164324         MOV     2(SP),@SWR      ;;SAVE NEW SWR
014614  062706  000006         11$:    ADD     #6,SP           ;;CLEAR UP STACK
014620  104401  001167         14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
014624  123727  001135  000001         CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
014632  001003                         BNE     15$             ;;BRANCH IF NOT
014634  012777  000100  164302         MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
014642  000002                 15$:    RTI                     ;;RETURN
014644  004737  015532         16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
014650  021627  000060                 CMP     (SP),#60        ;;CHAR < 0?
014654  002420                         BLT     18$             ;;BRANCH IF YES
014656  021627  000067                 CMP     (SP),#67        ;;CHAR > 7?
014662  003015                         BGT     18$             ;;BRANCH IF YES
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                                .DSABL  LSB
 4093
 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                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
 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    @$TKS           ;;WAIT FOR
 4106  014740  100375                        BPL     1$              ;;A CHARACTER
 4107  014742  117766  164200  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
 4108  014750  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4109  014756  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4110  014764  001013                        BNE     3$              ;;BRANCH IF NO
 4111  014766  105777  164152        2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 4112  014772  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
 4113  014774  117746  164146                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 4114  015000  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4115  015004  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4116  015010  001366                        BNE     2$              ;;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                                ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
 4130
 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                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
 4137  015064  122713  000177                  CMPB    #177,(R3)       ;;IS IT A RUBOUT
 4138  015070  001003                          BNE     3$              ;;SKIP IF NOT
 4139  015072  104401  001166                  TYPE    $QUES           ;;TYPE A '?'
 4140  015076  000763                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
 4141  015100  111337  015150          3$:     MOVB    (R3),9$         ;;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                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
 4146  015122  104401  001170                  TYPE    $LF             ;;TYPE A LINE FEED
 4147  015126  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 4148  015130  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
 4149  015132  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
 4150  015140  012766  015152  000004          MOV     #$TTYIN,4(SP)
 4151  015146  000002                          RTI                     ;;RETURN
 4152  015150     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
 4153  015151     000                          .BYTE   0               ;;TERMINATOR
 4154  015152  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
 4155  015162  052536  005015     000  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
 4156  015167     136  006507  000012  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
 4157  015174  005015  053523  020122  $MSWR:  .ASCIZ  <15><12>/SWR = /
 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                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
 4169                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
 4170                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
 4171
 4172  015216  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
 4173  015220  016666  000004  000002  MOV     4(SP),2(SP)     ;;INPUT NUMBER
 4174  015226  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 4175  015230  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 4176  015232  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 4177  015234  104411          1$:     RDLIN                   ;;READ AN ASCIZ LINE
 4178  015236  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
 4179  015240  005001                  CLR     R1              ;;CLEAR DATA WORD
 4180  015242  005002                  CLR     R2
 4181  015244  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
 4182  015246  001412                  BEQ     3$              ;;IF ZERO GET OUT
 4183  015250  006301                  ASL     R1              ;;*2
 4184  015252  006102                  ROL     R2
 4185  015254  006301                  ASL     R1              ;;*4
 4186  015256  006102                  ROL     R2
 4187  015260  006301                  ASL     R1              ;;*8
 4188  015262  006102                  ROL     R2
 4189  015264  042716  177770          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$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
 4193  015276  010166  000012                MOV     R1,12(SP)       ;;SAVE THE RESULT
 4194  015302  010237  015316                MOV     R2,$HIOCT
 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                     ;;RETURN
 4199  015316  000000                $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 4200                                 .SBTTL  TYPE ROUTINE
 4201
 4202                                 ;;**************************************************************
 4203                                 ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 4204                                 ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 4205                                 ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 4206                                 ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 4207                                 ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 4208                                 ;*
 4209                                 ;*CALL:
 4210                                 ;*1) USING A TRAP INSTRUCTION
 4211                                 ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
 4212                                 ;*OR
 4213                                 ;*      TYPE
 4214                                 ;*      MESADR
 4215                                 ;*
 4216
 4217  015320  105737  001157        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
 4218  015324  100002                        BPL     1$              ;;BR IF YES
 4219  015326  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
 4220  015330  000430                        BR      3$              ;;LEAVE
 4221  015332  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
 4222  015334  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
 4223  015340  122737  000001  001212        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 4224  015346  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
 4225  015350  132737  000100  001213        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
 4226  015356  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
 4227  015360  010037  015370                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
 4228  015364  004737  015610                JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
 4229  015370  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
 4230  015372  132737  000040  001213  62$:  BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
 4231  015400  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
 4232  015402  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
 4233  015404  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
 4234  015406  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
 4235  015410  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
 4236  015412  062716  000002          3$:   ADD     #2,(SP)         ;;ADJUST RETURN PC
 4237  015416  000002                        RTI                     ;;RETURN
 4238  015420  122716  000011          4$:   CMPB    #HT,(SP)        ;;BRANCH IF <HT>
 4239  015424  001430                        BEQ     8$
 4240  015426  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
 4241  015432  001006                        BNE     5$
 4242  015434  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
 4243  015436  104401                        TYPE                    ;;TYPE A CR AND LF
```

```
 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          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 4267  015536  100375                          BPL     $TYPEC
 4268  015540  116677  000002  163404          MOVB    2(SP),@$TPB     ;;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    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
 4294  015656  001425                          BEQ     3$              ;;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     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
4300  015704  105720              2$:       TSTB    (R0)+           ;;FIND END OF MESSAGE
4301  015706  001376                        BNE     2$
4302  015710  163700  001206                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
4303  015714  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
4304  015716  010037  001210                MOV     R0,$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,$TYPE        ;;CALL TYPE MACRO
4311  015756  000000              4$:       .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     $ENV            ;;RUNNING UNDER APT?
4316  015772  001413                        BEQ     12$             ;;IF NOT:  BR
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  $PWRDN: MOV     #$ILLUP,@#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,$SAVR6       ;;SAVE SP
4350  016110  012737  016122  000024        MOV     #$PWRUP,@#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  $PWRUP: MOV     #$ILLUP,@#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)+,@SWR      ;;POP STACK INTO @SWR
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     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
4369  016174  012737  000340  000026  MOV     #340,@#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          $ILLUP: 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 DECoDER
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     $TRPAD(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.
```

```
                                        ;       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
016310  020200  020040  020040  MESWCH: .ASCII  <200>#     NOTE:#
016316  047516  042524     072
016323     200  047506  020122          .ASCII  <200>#FOR THIS PROGRAM TO rUN PROPERLY, SWITCH#
016330  044124  051511  050040
016336  047522  051107  046501
016344  052040  020117  052522
016352  020116  051120  050117
016360  051105  054514  020054
016366  053523  052111  044103
016374  033600  020054  043117          .ASCII  <200>#7, OF THE VECTOR ADDRESS SWITCH PACK (E76),#
016402  052040  042510  053040
016410  041505  047524  020122
016416  042101  051104  051505
016424  020123  053523  052111
016432  044103  050040  041501
016440  020113  042450  033067
016446  026051
016450  046600  051525  020124          .ASCIZ  <200>#MUST BE ON. (M8200-YC BOARD)#<200>
016456  042502  047440  027116
016464  024040  034115  030062
016472  026460  041531  041040
016500  040517  042122  100051
016506     000
                                
016507     200  041125  051123  EM1:    .ASCIZ  <200>#UBSR ERROR#
016514  042440  051122  051117
016522     000
016523     200  047511  051123  EM2:    .ASCIZ  <200>#IOSR ERROR#
016530  042440  051122  051117
016536     000
016537     200  044523  042504  EM3:    .ASCIZ  <200>#SIDE 1 (IO TO UB) FAST PATH DATA ERROR#
016544  030440  024040  047511
016552  052040  020117  041125
016560  020051  040506  052123
016566  050040  052101  020110
016574  040504  040524  042440
016602  051122  051117     000
016607     200  044523  042504  EM4:    .ASCIZ  <200>#SIDE 2 (UB TO IO) FAST PATH DATA ERROR#
```

```
 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   $TSTNM,$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
```

```
ABASE = 170460          152#    236     277     373     374     375     376     377     378     379     380     381     382
                        383     389     390     391     392     393     394     395     396     397     398     399
ACDW1  = 000000         236     279
ACDW2  = 000000         236
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
aMSGLG= 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#
APTSPO= 000100          4225    4293    4334#
ASWREG= 000000           236     249
ATESTN= 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
```

```
                3146    3149    3170    3226    3294    3434    3591#
BIT0   = 000001  136#    2770    2878    2982    3090    3501    3508
BIT00  = 000001  126#     136
BIT01  = 000002  125#     135    3497
BIT02  = 000004  124#     134
BIT03  = 000010  123#     133
BIT04  = 000020  122#     132
BIT05  = 000040  121#     131
BIT06  = 000100  120#     130
BIT07  = 000200  119#     129
BIT08  = 000400  118#     128    3998
BIT09  = 001000  117#     127
BIT1   = 000002  135#    3058    3156    3501    3508
BIT10  = 002000  116#    3894
BIT11  = 004000  115#    4006
BIT12  = 010000  114#
BIT13  = 020000  113#    3901
BIT14  = 040000  112#    3984
BIT15  = 100000  111#    3555    3592
BIT2   = 000004  134#    3508
BIT3   = 000010  133#
BIT4   = 000020  132#
BIT5   = 000040  131#
BIT6   = 000100  130#    3002    3032    3034    3057    3106    3132    3134    3155
BIT7   = 000200  129#    2791    2813    2822    2849    2850    2895    2915    2922    2949    2950    3057    3155
BIT8   = 000400  128#
BIT9   = 001000  127#
BPTVEC= 000014  143#
BSEL0   001432  392#
BSEL1   001434  393#
BSEL2   001436  394#
BSEL3   001440  395#     587     663     735*    919     988*    1014*   1057*   1083*   1126*   1152*   1195*   1221*
                1264*   1290*   1333*   1359*   1402*   1428*   1471*   1497*   1552    1578    1619    1645    1686
                1712    1753    1779    1820    1846    1887    1913    1954    1980    2021    2047    2063*   2152
                2171*   2204*   2237*   2270*   2303*   2336*   2369*   2402*   2435*   2483    2515    2547    2579
                2611    2643    2675    2707    2739
BSEL4   001442  396#
BSEL5   001444  397#     533     753     818     895     953     2129    2770    2790    2791    2821    2822    2848
                2850    2982    3002    3004    3033    3034    3036    3059    3279
BSEL6   001446  398#
BSEL7   001450  399#
CKSWR  = 104407 3890    3916    3982    3983    4418#
CKVCT2  001454  402#
CKVECT  001452  401#
CR     = 000015  51#    4269    4279
CRLF   = 000200  52#     498    4240    4279
DDISP  = 177570  58#     218     432
DF0     017310  303     310     317     324     331     338     345     352     359    4522#
DH1     017102  301     308     315     322     329     336    4494#
DH11    017212  357    4507#
DH7     017142  343     350    4500#
DISPLA  001142  218#    432*    440*    3893*   4018*
DISPRE  000174   31#     440
DOIT    012166 3487    3497#
```

```
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   4472#
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
GTSWR  = 104406            492   4416#
HT     = 000011             49#   4238   4279
INNU     001460           404#   584*   621*   660*   697*   4517
IOTVEC=  000020           144#   417*   418*
ISSEX    012044          3416   3421   3437#  3441   3456
ISSUEC   011656           519    526    578    654    738    746    812    858    913    980    990   1016   1049
                         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
```

```
                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
MBSELO  001402   376#    472
MBSEL1  001404   377#   3497*   3501*   3508*
MBSEL2  001406   378#
MBSEL3  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*
MBSEL4  001412   380#
MBSEL5  001414   381#    556     776     838     879     939    2076    2878    2894    2895    2921    2922    2948
                2950    3090    3106    3108    3133    3134    3136    3157    3211
MBSEL6  001416   382#   3499*
MBSEL7  001420   383#
MCNT    001366   364#   3321
MESWCH  016310   503    4422#
MOCRAM  012632    37    3623#
MRCODE= ******  G   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*
PIRQ  = 177772   56#
PIRQVE= 000240  150#
PR0   = 000000   73#
PR1   = 000040   74#
PR2   = 000100   75#
PR3   = 000140   76#
PR4   = 000200   77#
PR5   = 000240   78#
PR6   = 000300   79#
PR7   = 000340   80#
PS    = 177776   53#     54
PSW   = 177776   54#
PWRVEC= 000024  145#    423*    424*   4340*   4341*   4350*   4356*   4368*   4369*
RANA    012626 3603    3611    3612*   3613*   3617#
RANB    012630 3606    3611*   3613    3614*   3618#
RAND    012600 3602    3611#
RBST  = 011770 3427#
RDCHR = 104410 4135    4419#
RDLIN = 104411 4177    4420#
RDOCT = 104412 3628    3636    3674    3688    3728    4421#
RECB    017316 4525#
RECBUF  017722 3188    3228    3260    3296    3604    3607    4525    4528#
RESVEC= 000010  140#
RE2B    017320 4526#
```

```
RE2BUF  020122          4526    4529#
RSTART  001706           448#
RTNAD   001712           449#   3378
R0      =%000000          61#    449*    452    453*    454    455*    456    457*    459    460*    471*    474    475*
                        2760*  2779*  2870*  2885*  2972*  2991*  3082*  3097*  3188*  3228*  3263*  3296*  3324*
                        3325*  3369*  3372   3403   3404*  3437*  3442   3472   3474*  3475*  3476*  3477*  3485*
                        3494*  3499   3523*  3540   3545*  3553   3574   3599*  3604   3607   3629*  3630*  3690*
                        3696   3717*  3721*  3822   3832*  3836   3852   3853   3866*  3930   3931*  3932*  3939*
                        3940*  3941*  3942*  3943*  3944   3949   3954*  3956*  3960   3962   4174   4178*  4181
                        4197*  4221   4222*  4227   4232   4235*  4287   4295*  4299   4300   4302*  4303*  4304
                        4326*  4342   4367*  4387   4388*  4389   4390*  4391*  4392*  4393*
R1      =%000001          62#    451*    456*    472*   3189*  3203*  3229*  3264*  3271*  3297*  3473   3478*  3479*
                        3480*  3481*  3482*  3493*  3637*  3643   3650   3665*  3711*  3712*  3713   3715   3719
                        3823   3836*  3837   3841   3865*  4175   4179*  4183*  4185*  4187*  4190*  4193   4196*
                        4288   4325*  4343   4366*
R2      =%000002          63#    458*    461*    473*    476*  3230*  3245*  3298*  3313*  3652*  3653   3675*  3677
                        3824   3835*  3839*  3842   3849*  3850*  3851   3856*  3864*  4176   4180*  4184*  4186*
                        4188*  4194   4195*  4344   4365*
R3      =%000003          64#   3661*  3662*  3663   3667   3765   3774*  3780*  3781*  3784*  3789*  3790*  3791
                        3800*  3825   3833*  3834*  3848*  3851*  3860*  3861*  3863*  4131   4132*  4133   4136*
                        4137   4141   4143   4145*  4147*  4345   4364*
R4      =%000004          65#   3522*  3561*  3766   3768*  3769*  3770*  3771   3772*  3786   3788*  3796*  3799*
                        4346   4363*
R5      =%000005          66#    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*  2065*  2071*  2093*
                        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*  2806*  2815*  2836*  2842*  2873*  2880*  2889*  2909*  2916*  2937*
                        2943*  2976*  2985*  2996*  3019*  3026*  3051*  3085*  3092*  3101*  3122*  3127*  3150*
                        3179*  3191*  3197*  3205*  3254*  3266*  3273*  3413*  3438*  3495*  3556*  3767   3773*
                        3775*  3777*  3778*  3779*  3780   3798*  3826   3828*  3830*  3837*  3841*  3856   3862*
                        4347   4362*
R6      =%000006          67#    411*    412*    413
R7      =%000007          68#
SENA      017314        4524#
SENBUF    017322        3185   3229   3263   3297   3599   4524   4527#
SP      =%000006          69#    415*    429*    437*    441  3354*  3361*  3403*  3433*  3437   3472*  3473*  3493
                        3494   3574*  3580*  3629   3637   3643*  3653*  3675   3690   3696*  3702*  3729   3757*
                        3758   3759   3760*  3765*  3766*  3767*  3773   3798   3799   3800   3801*  3802*  3822*
                        3823*  3824*  3825*  3826*  3827*  3828   3831*  3844   3846*  3848   3858   3860   3862
                        3863   3864   3865   3866   3868*  3869*  3898   3930*  3935*  3956   3960*  3989*  3992
                        3994   3995   4017   4019*  4036*  4037*  4038   4045*  4048*  4049*  4053*  4054*  4058
                        4061*  4065   4067   4069   4070*  4077   4079   4081*  4082   4084*  4085*  4086*  4087*
                        4088*  4103*  4104*  4107*  4108*  4109   4113*  4114*  4115   4118   4120   4122*  4131*
                        4136   4147   4148*  4149*  4150*  4172*  4173*  4174*  4175*  4176*  4178   4181*  4189*
                        4190   4192   4193*  4195   4196   4197   4221*  4222   4232*  4234   4235   4236*  4238
                        4240   4242   4248   4250*  4252*  4260*  4264   4268   4269   4273   4287*  4288*  4295
```

```
                    4296*  4307   4308*  4309*  4319   4320*  4325   4326   4342*  4343*  4344*  4345*  4346*
                    4347*  4348*  4349   4357*  4361   4362   4363   4364   4365   4366   4367   4387*  4388
                    4398*  4399*
STACK = 001100        44#    415    3433
START   001464        35     408#
STKLMT= 177774        55#
SWR     001140       217#    413    431*   433    439*   446*   490    3894   3901   3913   3984   3998   4000
                    4006   4032   4069*  4348   4361*
SWREG   000176        32#    439    490    4032   4045
SW0   = 000001       108#
SW00  = 000001        98#    108
SW01  = 000002        97#    107
SW02  = 000004        96#    106
SW03  = 000010        95#    105
SW04  = 000020        94#    104
SW05  = 000040        93#    103
SW06  = 000100        92#    102
SW07  = 000200        91#    101
SW08  = 000400        90#    100
SW09  = 001000        89#     99
SW1   = 000002       107#
SW10  = 002000        88#
SW11  = 004000        87#
SW12  = 010000        86#
SW13  = 020000        85#
SW14  = 040000        84#
SW15  = 100000        83#
SW2   = 000004       106#
SW3   = 000010       105#
SW4   = 000020       104#
SW5   = 000040       103#
SW6   = 000100       102#
SW7   = 000200       101#
SW8   = 000400       100#
SW9   = 001000        99#
TBITVE= 000014       141#
TIME    012244      3414*  3425*  3512*  3514#
TKVEC = 000060       148#
TPVEC = 000064       149#
TRAPVE= 000034       147#    421*   422*
TRTVEC= 000014       142#
TST1    002204       509#
TST10   003106       768     789    808    830    842    855#
TST11   003340       893     908    931    951    957    977#
TST12   003464      1029    1046#
TST13   003610      1098    1115#
TST14   003734      1167    1184#
TST15   004060      1236    1253#
TST16   004204      1305    1322#
TST17   004330      1374    1391#
TST2    002316       538     547#
TST20   004454      1443    1460#
TST21   004600      1512    1529#
TST22   004722      1579    1596#
```

```
TST23    005044            1646    1663#
TST24    005166            1713    1730#
TST25    005310            1780    1797#
TST26    005432            1847    1864#
TST27    005554            1914    1931#
TST3     002366             561     574#
TST30    005676            1981    1998#
TST31    006020            2048    2060#
TST32    006130            2090    2099    2115#
TST33    006240            2143    2153    2168#
TST34    006314            2187    2201#
TST35    006370            2220    2234#
TST36    006444            2253    2267#
TST37    006520            2286    2300#
TST4     002440             589     613#
TST40    006574            2319    2333#
TST41    006650            2352    2366#
TST42    006724            2385    2399#
TST43    007000            2418    2432#
TST44    007054            2451    2466#
TST45    007130            2485    2498#
TST46    007204            2517    2530#
TST47    007260            2549    2562#
TST5     002510             626     650#
TST50    007334            2581    2594#
TST51    007410            2613    2626#
TST52    007464            2645    2658#
TST53    007540            2677    2690#
TST54    007614            2709    2722#
TST55    007670            2741    2756#
TST56    010130            2802    2833    2866#
TST57    010356            2906    2933    2967#
TST6     002562             665     689#
TST60    010630            3015    3047    3077#
TST61    011066            3119    3147    3175#
TST62    011256            3227    3244    3251#
TST63    011434            3295    3312    3319#
TST7     002632             702     726#
TYPDS  = 104405            3356    3362    4414#
TYPE   = 104401             484     503    3350  3357  3364  3429  3564  3568  3576  3624  3632  3639  3645
                           3655    3670    3684  3692  3698  3705  3724  3792  3867  3896  3904  3929  3946
                           3948    3951    3953  3957  3964  4043  4044  4047  4060  4071  4090  4139  4142
                           4146    4243    4370  4410#
TYPOC  = 104402            3575    3581    3644  3654  3697  3703  3937  3961  4046  4411#
TYPON  = 104404            4413#
TYPOS  = 104403            4412#
UCODEE= ******  G             1#    3540    3553
VAR1     001364             363#    465
VAR2     001370             365#   3325
$APTHD   001000             177     183#
$ASTAT= ****** U           4313    4328
$ATYC    015626            4284    4286#
$ATY1    015602            4282#
$ATY3    015610            4228    4283#
```

```
$ATY4   015620  3909  4285#
$AUTOB  001134   214#   494*  4040  4161
$BASE   001246   277#   449
$BDADR  001122   209#
$BDDAT  001126   211#   533*  534*  537   556*  557*  560   586*  587*  588   623*  624*  625
                 662*  663*  664   699*  700*  701   753*  754*  757   776*  779   796*  818*
                 820   838*  841   879*  882   895*  898   919*  939*  941   953*  956   1002*
                1003  1028*  1071*  1072  1097*  1140*  1141  1166*  1209*  1210  1235*  1278*  1279
                1304*  1347*  1348  1373*  1416*  1417  1442*  1485*  1486  1511*  1552*  1553  1578*
                1619*  1620  1645*  1686*  1687  1712*  1753*  1754  1779*  1820*  1821  1846*  1887*
                1888  1913*  1954*  1955  1980*  2021*  2022  2047*  2076*  2078  2098*  2129*  2131
                2152*  2185*  2186  2218*  2219  2251*  2252  2284*  2285  2317*  2318  2350*  2351
                2383*  2384  2416*  2417  2449*  2450  2483*  2484  2515*  2516  2547*  2548  2579*
                2580  2611*  2612  2643*  2644  2675*  2676  2707*  2708  2739*  2740  2790*  2821*
                2848*  2894*  2921*  2948*  3004*  3033*  3036*  3059*  3060  3108*  3133*  3136*  3157*
                3158  3210*  3211*  3213  3233*  3278*  3279*  3281  3301*  3443*  3444*  4515  4517
                4519
$BELL   001162   228#  3896  3921
$CDWI   001252   279#
$CHARC  015576  4245*  4255*  4262  4271*  4276#
$CKSWR  014442  4032#  4418
$CMTAG  001100   197#   410  411  419  425  426
$CM3  =  000000   227#
$CNTLG  015167  4043  4156#
$CNTLU  015162  4060  4155#
$CPUOP  001220   251#
$CRLF   001167   230#  3904  3921  3929  3948  3953  3957  4071  4155  4244  4279
$DBLK   013716  3833  3867  3875#
$DEVCT  001202   242#
$DEVM   001250   278#
$DOAGN  011646  3344  3370  3376#
$DTBL   013706  3836  3871#
$ENDAD  011636   163  3372#
$ENDCT  011522  3346#
$ENULL  011652  3379#
$ENV    001212   247#   488  3906  4223  4291  4315
$ENVM   001213   248#   444  4225  4230  4293
$EOP    011466  3336#
$EOPCT  011514  3343#  3347
$ERFLG  001103   200#  3891*  3921  3973  4002  4004*  4022
$ERMAX  001115   206#  4022
$ERROR  013726   419  3889#
$ERrPC  001116   207#  3898*  3899*  3900  3921  3935  4515  4517  4519
$ERRTB  001254   296#  3943
$ERrTY  014074  3903  3928#
$ERTTL  001112   204#  3897*  3921
$ETABL  001212   246#
$ETEND  001254   189  280#
$FATAL  001174   239#  4319*
$FFLG   016046  4282*  4285*  4313  4322*  4330#
$FILLC  001156   225#  4248  4279
$FILLS  001155   224#  4279
$GDADR  001120   208#
$GDDAT  001124   210#  535*  537  558*  560  585*  588  622*  625  661*  664  698*  701
```

```
$OCNT   013476          3764*  3793*  3806#
$OMODE  013500          3759*  3763*  3768   3771*  3782*  3808#
$OVER   014424          3985   3996   4001   4012   4018#
$PASS   001200           241#   443*  3340*  3341*  3354   3379   4008   4022
$PASTM  001006           187#
$POWER  016216          4371   4376#
$PWRDN  016050           423   4340#  4368
$PWRMG  016204          4371#
$PWRUP  016122          4350   4356#
$QUES   001166           229#  3921   4090   4139   4155   4279
$RDCHR  014724          4103#  4419
$RDDEC= ******  U       4422
$RDLIN  015044          4131#  4420
$RDOCT  015216          4172#  4421
$RDSZ = 000010          4124#
$RTNAD  011650          3378#
$R2A  = ******  U       4422
$SAVRE= ******  U       4422
$SAVR6  016214          4349*  4357   4358*  4359*  4375#
$SCOPE  014230           417   3961#
$SETUP= 000117           407#   416    417    419    421    423    425    426    484    485   3338   3890   3916
                        3918   3982   4027   4161
$STUP = 177777           407#
$SVLAD  014406          3993   4015#
$SVPC = 000224           161#   166
$SWR  = 166400             1#    12     18     19     20     21     22     23    227    228    425    426    427
                         510    548    575    614    651    690    727    856    978   1047   1116   1185   1254
                        1323   1392   1461   1530   1597   1664   1731   1798   1865   1932   1999   2061   2116
                        2169   2202   2235   2268   2301   2334   2367   2400   2433   2467   2499   2531   2563
                        2595   2627   2659   2691   2723   2757   2867   2968   3078   3176   3252   3320   3333
                        3339   3371   3377   3379   3882   3883   3884   3885   3886   3894   3901   3913   3917
                        3921   3974   3975   3976   3977   3984   3996   3998   3999   4002   4003   4004   4005
                        4006   4018   4021   4372
$SWREG  001214           249#   446
$SWRMK= 000000            23     24   3977   3978   4000
$TESTN  001176           240#  4016*
$TIMES  001160           227#   425*   510*   548*  2757*  2867*  2968*  3078*  3339*  4005*  4011   4014*  4021
$TKB    001146           220#  3661   3711   4025   4036   4053   4107   4113
$TKs    001144           219#  3659   3709   4025   4034   4050   4074*  4105   4111
$TN   = 000064            12#   155*   506    510#   538    544    548#   561    571    575#   589    610    614#
                         626    647    651#   665    686    690#   702    723    727#   768    789    808    830
                         842    852    856#   893    908    931    951    957    969    978#  1029   1038   1047#
                        1098   1107   1116#  1167   1176   1185#  1236   1245   1254#  1305   1314   1323#  1374
                        1383   1392#  1443   1452   1461#  1512   1521   1530#  1579   1588   1597#  1646   1655
                        1664#  1713   1722   1731#  1780   1789   1798#  1847   1856   1865#  1914   1923   1932#
                        1981   1990   1999#  2048   2057   2061#  2090   2099   2112   2116#  2143   2153   2165
                        2169#  2187   2198   2202#  2220   2231   2235#  2253   2264   2268#  2286   2297   2301#
                        2319   2330   2334#  2352   2363   2367#  2385   2396   2400#  2418   2429   2433#  2451
                        2463   2467#  2485   2495   2499#  2517   2527   2531#  2549   2559   2563#  2581   2591
                        2595#  2613   2623   2627#  2645   2655   2659#  2677   2687   2691#  2709   2719   2723#
                        2741   2753   2757#  2802   2833   2863   2867#  2906   2933   2964   2968#  3015   3047
                        3074   3078#  3119   3147   3172   3176#  3227   3244   3248   3252#  3295   3312   3316
                        3320#
$TPB    001152           222#  4268*  4279
```

```
$TPFLG   001157            226#  4217  4279
$TPS     001150            221#  4266  4279
$TRAP    016226            421   4387#
$TRAP2   016250           4398#  4409
$TRP   = 000013           4402#  4411#  4412#  4413#  4414#  4415#  4416  4417#  4418  4419#  4420#  4421#  4422#
$TRPAD   016262           4392   4409#
$TSTM    001004            186#
$TSTNM   001102            199#   513#  3338#  3893  3921  3973  4000  4015#  4016  4018  4022  4519
$TTYIN   015152           4132   4133  4150  4154#
$TYPBN=  ****** U         4415
$TYPDS   013502           3821#  4414
$TYPE    015320           4217#  4310  4402  4410
$TYPEC   015532           4076   4247  4254  4261  4266#  4267
$TYPEX   015600           4272   4274  4277#
$TYPOC   013300           3762#  4411
$TYpON   013314           3761   3764#  4413
$TYPOS   013254           3757#  4412
$UNIT    001204            243#
$UNItM   001010            188#
$USWR    001216            250#
$VECT1   001242            275#
$VECT2   001244            276#
$XTSTR   014244           3987#
$$GEt4=  000000           3371#
$OFILL   013477           3758*  3762*  3772  3807#
$40CAT=  ****** U         3903   3984
.        020322             26#    30#    34#    36#    38#   161   162#   164#   166#   167#   173   174#   176#
                           178#   196#   232   414   426  3353#  3360#  3367#  3379  3380#  3427  3432#  3560
                          3567#  3627#  3635#  3648#  3701#  3727#  3875#  3921  3967#  4021  4022  4025  4154#
                          4155   4161  4279  4331#  4352  4374  4513#  4527#  4528#  4529#
.MD.     000001            521#   523   528#   530   551#   553   580#   582   617#   619   656#   658   693#
                           695   730#   732   740#   742   748#   750   772#   774   792#   794   814#   816
                           834#   836   860#   862   868#   870   874#   876   915#   917   935#   937   982#
                           984   992#   994   997#   999  1018#  1020  1023#  1025  1051#  1053  1061#  1063
                          1066#  1068  1087#  1089  1092#  1094  1120#  1122  1130#  1132  1135#  1137  1156#
                          1158  1161#  1163  1189#  1191  1199#  1201  1204#  1206  1225#  1227  1230#  1232
                          1258#  1260  1268#  1270  1273#  1275  1294#  1296  1299#  1301  1327#  1329  1337#
                          1339  1342#  1344  1363#  1365  1368#  1370  1396#  1398  1406#  1408  1411#  1413
                          1432#  1434  1437#  1439  1465#  1467  1475#  1477  1480#  1482  1501#  1503  1506#
                          1508  1532#  1534  1540#  1542  1547#  1549  1566#  1568  1573#  1575  1599#  1601
                          1607#  1609  1614#  1616  1633#  1635  1640#  1642  1666#  1668  1674#  1676  1681#
                          1683  1700#  1702  1707#  1709  1733#  1735  1741#  1743  1748#  1750  1767#  1769
                          1774#  1776  1800#  1802  1808#  1810  1815#  1817  1834#  1836  1841#  1843  1867#
                          1869  1875#  1877  1882#  1884  1901#  1903  1908#  1910  1934#  1936  1942#  1944
                          1949#  1951  1968#  1970  1975#  1977  2001#  2003  2009#  2011  2016#  2018  2035#
                          2037  2042#  2044  2067#  2069  2072#  2074  2094#  2096  2120#  2122  2125#  2127
                          2148#  2150  2175#  2177  2180#  2182  2208#  2210  2213#  2215  2241#  2243  2246#
                          2248  2274#  2276  2279#  2281  2307#  2309  2312#  2314  2340#  2342  2345#  2347
                          2373#  2375  2378#  2380  2406#  2408  2411#  2413  2439#  2441  2444#  2446  2471#
                          2473  2478#  2480  2503#  2505  2510#  2512  2535#  2537  2542#  2544  2567#  2569
                          2574#  2576  2599#  2601  2606#  2608  2631#  2633  2638#  2640  2663#  2665  2670#
                          2672  2695#  2697  2702#  2704  2727#  2729  2734#  2736  2766#  2768  2775#  2777
                          2786#  2788  2808#  2810  2817#  2819  2837#  2839  2844#  2846  2874#  2876  2881#
                          2883   2890#  2892  2910#  2912  2917#  2919  2939#  2941  2944#  2946  2978#  2980
```

```
COMMEN   151#
DFC      156#   303   310   317   324   331   338   345   352   359
DXFRF    967#   968  1037  1106  1175  1244  1313  1382  1451  1520  1587  1654  1721  1788  1855
        1922  1989
ECB      362#   539   565   590   606   627   643   666   682   703   719   759   765   781   786
         800   805   822   827   843   848   884   890   900   905   923   928   943   948   958
         963  1005  1010  1030  1034  1074  1079  1099  1103  1143  1148  1168  1172  1212  1217
        1237  1241  1281  1286  1306  1310  1350  1355  1375  1379  1419  1424  1444  1448  1488
        1493  1513  1517  1555  1560  1580  1584  1622  1627  1647  1651  1689  1694  1714  1718
        1756  1761  1781  1785  1823  1828  1848  1852  1890  1895  1915  1919  1957  1962  1982
        1986  2024  2029  2049  2053  2080  2087  2103  2108  2133  2140  2158  2189  2194  2222
        2227  2255  2260  2288  2293  2321  2326  2354  2359  2387  2392  2420  2425  2453  2458
        2486  2491  2518  2523  2550  2555  2582  2587  2614  2619  2646  2651  2678  2683  2710
        2715  2742  2747  2793  2798  2824  2829  2852  2858  2897  2902  2924  2929  2952  2958
        3005  3011  3037  3043  3062  3068  3109  3115  3137  3143  3160  3166  3215  3223  3235
        3241  3283  3291  3303  3309  3445  3452
ENDCOM   151#
ENDPAS  3329#  3348
ERROR     45#   542   563   593   630   669   706   762   784   803   825   846   887   903   926
         946   961  1008  1033  1077  1102  1146  1171  1215  1240  1284  1309  1353  1378  1422
        1447  1491  1516  1558  1583  1625  1650  1692  1717  1759  1784  1826  1851  1893  1918
        1960  1985  2027  2052  2083  2106  2136  2161  2192  2225  2258  2291  2324  2357  2390
        2423  2456  2489  2521  2553  2585  2617  2649  2681  2713  2745  2796  2827  2855  2900
        2927  2955  3008  3040  3065  3112  3140  3163  3218  3238  3286  3306  3448
ESCAPE   151#
GETPRI   151#
GETSWR   151#   485#
INSTR2  3876#  3918
IN1367   569#   570   609   646   685
MCMD     361#   517   525   550   577   616   653   692   729   737   745   770   791   810   832
         857   867   873   911   933   979   989   996  1015  1022  1048  1058  1065  1084  1091
        1117  1127  1134  1153  1160  1186  1196  1203  1222  1229  1255  1265  1272  1291  1298
        1324  1334  1341  1360  1367  1393  1403  1410  1429  1436  1462  1472  1479  1498  1505
        1531  1539  1544  1565  1570  1598  1606  1611  1632  1637  1665  1673  1678  1699  1704
        1732  1740  1745  1766  1771  1799  1807  1812  1833  1838  1866  1874  1879  1900  1905
        1933  1941  1946  1967  1972  2000  2008  2013  2034  2039  2064  2071  2093  2119  2124
        2144  2172  2179  2205  2212  2238  2245  2271  2278  2304  2311  2337  2344  2370  2377
        2403  2410  2436  2443  2470  2475  2502  2507  2534  2539  2566  2571  2598  2603  2630
        2635  2662  2667  2694  2699  2726  2731  2763  2772  2783  2804  2814  2835  2841  2873
        2880  2889  2908  2916  2935  2943  2974  2984  2995  3017  3025  3050  3084  3092  3101
        3121  3127  3150  3178  3190  3197  3205  3253  3266  3273
MULT     151#
NEWTST   151#   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
POP      151#  3862  4195  4325  4326  4361  4362
PUSH     151#  3821  4174  4286  4288  4309  4342  4348
REPORT   151#
SCOPE     46#   547   574   613   650   689   726   855   977  1046  1115  1184  1253  1322  1391
        1460  1529  1596  1663  1730  1797  1864  1931  1998  2060  2115  2168  2201  2234  2267
        2300  2333  2366  2399  2432  2466  2498  2530  2562  2594  2626  2658  2690  2722  2756
        2866  2967  3077  3175  3251  3319
SETPRI   151#
```

```
SETTRA   4402#  4411  4412  4413  4414  4416  4418  4419  4420  4421
SETUP     151#   408
SIDA     2164#  2165  2198  2231  2264  2297  2330  2363  2396  2429  2463  2495  2527  2559  2591
         2623   2655  2687  2719
SKIP      151#   538   561   589   626   665   702   768   789   808   830   842   893   908   931
          951    957  1029  1098  1167  1236  1305  1374  1443  1512  1579  1646  1713  1780  1847
         1914   1981  2048  2090  2099  2143  2153  2187  2220  2253  2286  2319  2352  2385  2418
         2451   2485  2517  2549  2581  2613  2645  2677  2709  2741  2802  2833  2906  2933  3015
         3047   3119  3147  3227  3244  3295  3312
SLASH     151#
SOVFL    2963#  3073
SPACE     151#
STARS     151#   159   170   172   179   192   232   235   506   508   544   546   571   573   610
          612    647   649   686   688   723   725   852   854   969   976  1038  1045  1107  1114
         1176   1183  1245  1252  1314  1321  1383  1390  1452  1459  1521  1528  1588  1595  1655
         1662   1722  1729  1789  1796  1856  1863  1923  1930  1990  1997  2057  2059  2112  2114
         2165   2167  2198  2200  2231  2233  2264  2266  2297  2299  2330  2332  2363  2365  2396
         2398   2429  2431  2463  2465  2495  2497  2527  2529  2559  2561  2591  2593  2623  2625
         2655   2657  2687  2689  2719  2721  2753  2755  2863  2865  2964  2966  3074  3076  3172
         3174   3248  3250  3316  3318  3331  3734  3811  3878  3923  3970  4024  4027  4095  4124
         4164   4202  4281  4338  4354  4381
SWRSU     151#   427#
S78F     2752#  2862
TRMTRP   4402#
TYPBIN    151#
TYPDEC    151#  3354  3361
TYPNAM      1#   151#   480
TYPNUM    151#
TYPOCS      1#   151#
TYPOCT    151#  3573  3580  3643  3653  3696  3702  3935  3959  4045
tYPTXT    151#  3350  3357  3364  3429  3564  3568  3576  3623  3631  3638  3645  3655  3670  3683
         3691   3698  3704  3723
ZZ1       969#   971  1038#  1040  1107#  1109  1176#  1178  1245#  1247  1314#  1316  1383#  1385  1452#
         1454   1521#  1523  1588#  1590  1655#  1657  1722#  1724  1789#  1791  1856#  1858  1923#  1925
         1990#  1992
$$CmRE    190#
$$CMTM    190#
$$EsCA    151#
$$NEWT    151#   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
$$SET    4402#  4411  4412  4413  4414  4416  4418  4419  4420  4421
$$SETM    443#
$$SkIP    151#   538   561   589   626   665   702   768   789   808   830   842   893   908   931
          951    957  1029  1098  1167  1236  1305  1374  1443  1512  1579  1646  1713  1780  1847
         1914   1981  2048  2090  2099  2143  2153  2187  2220  2253  2286  2319  2352  2385  2418
         2451   2485  2517  2549  2581  2613  2645  2677  2709  2741  2802  2833  2906  2933  3015
         3047   3119  3147  3227  3244  3295  3312
.EQUAT      1#    41
.HEADE      1#     2
.SETTR      1#
.SETUP      1#   407
.SWRHI      1#    14
```

```
.SWRLO      24#
.TRMTR       1#
.$ACT1       1#     157
.$APTB     233#
.$APTH       1#     168
.$APTY       1#    4279
.$CATC       1#      24
.$CMTA       1#     190
.$EOP        1#    3329
.$ERRO       1#    3876
.$ERRT       1#    3921
.$POWE       1#    4336
.$RDOC       1#    4162
.$READ       1#    4022
.$SCOP       1#    3968
.$TRAP       1#    4379
.$TYPD       1#    3809
.$TYpE       1#    4200
.$TYPO       1#    3732
```

```
          4269  4273  4291
DEC        461   476  2779  2885  2991  3097  3203  3245  3271  3313  3342  3939
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
          4010  4087  4321  4359
INCB      3561  3891  4015  4275
IOT         46
JMP         35    37    39  3326  3377  3435  3584
JSR        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  3903  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  2760  2782  2867  2870  2888
          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
          4387  4388  4392  4398  4399
MOVB       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
```

```
                4141  4181  4232  4260  4268  4282  4283  4285  4390
NEG             3769  3830
NOP              509  3337  3373  3374  3375  3498  3500  3502  3503  3504  3505  3506  3507
RESET            478  3371  3591
ROL             3775  3777  3778  3779  3781  4184  4186  4188
RTI              438  3513  3803  3870  3920  4020  4075  4123  4151  4198  4237  4372  4400
RTS             3438  3495  3509  3556  3594  3609  3615  3958  4277  4327  4393
SUB             3721  3837  3899  4302
TRAP            4402  4411  4412  4413  4414  4416  4418  4419  4420  4421
TST              486  3572  3786  3842  3852  3913  3962  3991  4008  4067  4092  4192  4234  4242  4264
                4297  4315  4317  4389
TSTB            3405  3415  3420  3423  3659  3709  3844  3858  4002  4034  4050  4105  4111  4217  4266
                4289  4300  4313
.ASCII           229   230  4422  4424  4431
.ASCIZ           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    21    23    24    33    45   137   151   160   164   166   171   173   180   193
                 197   199   227   228   229   233   236   258   266   269   272   275   276   277   278
                 279   282   407   415   416   419   421   423   425   426   427   448   484   490   496
                 498   507   508   509   510   511   512   520   521   522   523   524   527   528   530
                 531   539   545   546   547   548   549   551   553   554   562   572   573   574   575
                 579   580   581   582   583   590   611   612   613   614   617   618   619   620   627
                 648   649   650   651   655   656   657   658   659   666   687   688   689   690   693
                 694   695   696   703   724   725   726   727   730   731   732   733   739   740   741
                 742   743   747   748   750   751   769   772   774   775   790   792   793   794   795
                 809   813   814   816   817   831   834   836   837   843   853   854   855   856   859
                 860   861   862   863   868   869   870   871   874   876   877   894   909   914   915
                 916   917   918   932   935   937   938   952   958   970   971   976   977   978   981
                 982   984   985   991   992   993   994   995   997   998   999  1000  1009  1017  1018
                1019  1020  1021  1023  1024  1025  1026  1030  1034  1039  1040  1045  1046  1047  1050
                1051  1053  1054  1060  1061  1062  1063  1064  1066  1067  1068  1069  1078  1086  1087
                1088  1089  1090  1092  1093  1094  1095  1099  1103  1108  1109  1114  1115  1116  1119
                1120  1122  1123  1129  1130  1131  1132  1133  1135  1136  1137  1138  1147  1155  1156
                1157  1158  1159  1161  1162  1163  1164  1168  1172  1177  1178  1183  1184  1185  1188
                1189  1191  1192  1198  1199  1200  1201  1202  1204  1205  1206  1207  1216  1224  1225
                1226  1227  1228  1230  1231  1232  1233  1237  1241  1246  1247  1252  1253  1254  1257
                1258  1260  1261  1267  1268  1269  1270  1271  1273  1274  1275  1276  1285  1293  1294
                1295  1296  1297  1299  1300  1301  1302  1306  1310  1315  1316  1321  1322  1323  1326
                1327  1329  1330  1336  1337  1338  1339  1340  1342  1343  1344  1345  1354  1362  1363
                1364  1365  1366  1368  1369  1370  1371  1375  1379  1384  1385  1390  1391  1392  1395
                1396  1398  1399  1405  1406  1407  1408  1409  1411  1412  1413  1414  1423  1431  1432
                1433  1434  1435  1437  1438  1439  1440  1444  1448  1453  1454  1459  1460  1461  1464
                1465  1467  1468  1474  1475  1476  1477  1478  1480  1481  1482  1483  1492  1500  1501
                1502  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	2232	2233	2234	2235	2240	2241	2242	2243
2244	2246	2247	2248	2249	2254	2259	2265	2266	2267	2268	2273	2274	2275	2276
2277	2279	2280	2281	2282	2287	2292	2298	2299	2300	2301	2306	2307	2308	2309
2310	2312	2313	2314	2315	2320	2325	2331	2332	2333	2334	2339	2340	2341	2342
2343	2345	2346	2347	2348	2353	2358	2364	2365	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	2497	2498	2499	2503	2504	2505	2506
2509	2510	2511	2512	2513	2518	2522	2528	2529	2530	2531	2535	2536	2537	2538
2541	2542	2543	2544	2545	2550	2554	2560	2561	2562	2563	2567	2568	2569	2570
2573	2574	2575	2576	2577	2582	2586	2592	2593	2594	2595	2599	2600	2601	2602
2605	2606	2607	2608	2609	2614	2618	2624	2625	2626	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	2682	2688	2689	2690	2691	2695	2696	2697	2698
2701	2702	2703	2704	2705	2710	2714	2720	2721	2722	2723	2727	2728	2729	2730
2733	2734	2735	2736	2737	2742	2746	2754	2755	2756	2757	2758	2765	2766	2768
2769	2774	2775	2776	2777	2778	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	2934	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	3176	3180	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	3567	3571	3579	3627	3635	3642	3648	3658	3673	3687
3695	3701	3708	3727	3735	3812	3879	3882	3891	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
.EQUIV    45      46      54      99     100     101     102     103     104     105     106     107     108     127     128
         129     130     131     132     133     134     135     136
.EVEN     236     498    3353    3360    3367    3380    3432    3567    3571    3579    3627    3635    3642    3648    3658
         3673    3687    3695    3701    3708    3727    3967    4331    4378    4513
.GLOBL       1
.IF          3      21      22      23      24      33      43     109     137     159     162     164     170     172     179
           192     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     976     978     979     981     983     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    2027    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  2165  2167  2169  2172  2174  2176  2177  2178  2179  2180  2181
2182  2183  2187  2188  2198  2200  2202  2205  2207  2209  2210  2211  2212  2213  2214
2215  2216  2220  2221  2231  2233  2235  2238  2240  2242  2243  2244  2245  2246  2247
2248  2249  2253  2254  2264  2266  2268  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  2330  2332  2334  2337  2339  2341  2342  2343  2344  2345  2346
2347  2348  2352  2353  2363  2365  2367  2370  2372  2374  2375  2376  2377  2378  2379
2380  2381  2385  2386  2396  2398  2400  2403  2405  2407  2408  2409  2410  2411  2412
2413  2414  2418  2419  2429  2431  2433  2436  2438  2440  2441  2442  2443  2444  2445
2446  2447  2451  2452  2463  2465  2467  2470  2471  2472  2473  2474  2475  2477  2479
2480  2481  2485  2486  2496  2497  2499  2502  2503  2504  2505  2506  2507  2509  2511
2512  2513  2517  2518  2527  2529  2531  2534  2535  2536  2537  2538  2539  2541  2543
2544  2545  2549  2550  2559  2561  2563  2566  2567  2568  2569  2570  2571  2573  2575
2576  2577  2581  2582  2591  2593  2595  2598  2599  2600  2601  2602  2603  2605  2607
2608  2609  2613  2614  2623  2625  2627  2630  2631  2632  2633  2634  2635  2637  2639
2640  2641  2645  2646  2655  2657  2659  2662  2663  2664  2665  2666  2667  2669  2671
2672  2673  2677  2678  2687  2689  2691  2694  2695  2696  2697  2698  2699  2701  2703
2704  2705  2709  2710  2719  2721  2723  2726  2727  2728  2729  2730  2731  2733  2735
2736  2737  2741  2742  2753  2755  2757  2758  2763  2765  2767  2768  2769  2772  2774
2776  2777  2778  2782  2783  2785  2787  2788  2789  2802  2805  2807  2809  2810  2811
2814  2816  2818  2819  2820  2833  2836  2837  2838  2839  2840  2841  2843  2845  2846
2847  2863  2865  2867  2868  2873  2874  2875  2876  2877  2880  2881  2882  2883  2884
2888  2889  2890  2891  2892  2893  2906  2909  2910  2911  2912  2913  2916  2917  2918
2919  2920  2933  2936  2938  2940  2941  2942  2943  2944  2945  2946  2947  2964  2966
2968  2969  2975  2977  2979  2980  2981  2984  2986  2988  2989  2990  2993  2995  2997
2999  3000  3001  3015  3018  3020  3022  3023  3024  3025  3027  3029  3030  3031  3047
3050  3052  3054  3055  3056  3074  3076  3078  3079  3085  3086  3087  3088  3089  3092
3093  3094  3095  3096  3099  3101  3102  3103  3104  3105  3119  3122  3123  3124  3125
3126  3127  3128  3129  3130  3131  3147  3150  3151  3152  3153  3154  3172  3174  3176
3178  3180  3182  3184  3185  3191  3192  3193  3194  3195  3197  3198  3199  3200  3201
3205  3206  3207  3208  3209  3227  3244  3248  3250  3252  3253  3255  3257  3259  3260
3266  3267  3268  3269  3270  3273  3274  3275  3276  3277  3295  3312  3316  3318  3320
3331  3332  3333  3334  3335  3337  3343  3346  3348  3352  3359  3366  3369  3371  3377
3379  3380  3431  3566  3570  3578  3626  3634  3641  3647  3657  3672  3686  3694  3700
3707  3726  3734  3811  3878  3881  3891  3894  3901  3903  3904  3906  3913  3917  3918
3921  3923  3938  3954  3970  3973  3977  3983  3984  3996  3998  3999  4000  4002  4003
4004  4006  4008  4016  4018  4020  4021  4022  4024  4026  4027  4028  4056  4095  4096
4124  4132  4133  4137  4138  4154  4155  4161  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
.IFF    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   562   572   573   574   575   577   580   590   611   612
 613   614   617   627   648   649   650   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   958   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
```

```
                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  2932  2952  2955  2958  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
.MACRO            24   156   190   361   362   443   569   967   969  1038  1107  1176  1245  1314  1383
                1452  1521  1588  1655  1722  1789  1856  1923  1990  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   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   764   765   768   781   784   786   789   800   803   805   808   822   825
                 827   830   845   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  2932  2952  2955  2958  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     282
.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    2569    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

```
 *DRLPN,DRLPN/CRF/SOL=DRLPN
 RUN-TIME:  36 26 4 SECONDS
  CORE USED:   28K
EOF1DRLPNASEQ              00010000      780223            PDP10 411
```