```
                        .REM "
```

IDENTIFICATION

PRODUCT CODE: AC-E992B-MC

PRODUCT NAME: CXKUAB0 KUV11-A MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

KUA IS AN NBKMOD FOR THE KUV11-AA LSI-11 WRITABLE CONTROL STORE OPTION. IT IS USED IN CONJUCTION WITH THE CPB (EIS EXERCISER) AND FPA (FIS EXERCISER) DECX MODULES TO EXERCISE THE KUV11-AA IN ADDRESS MODE 1 OR 3. THE KUA MODULE RUNS FIRST, RUNS ONLY ONCE, AND WITH THE T-BIT OFF. ITS PURPOSE IS TO RUN A MEMORY TEST ON THE KUV11-AA RAM AND THEN TO LOAD THE RAM WITH THE APPROPRIATE MICRO-CODE FOR THE CPB AND FPA MODULES TO RUN. THUS, CPB AND FPA SHOULD BE CONFIGURED TO RUN ON AN LSI-11, IN ORDER TO FULLY EXERCISE THE KUV11-AA.
ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TERMINAL.

2. REQUIREMENTS

HARDWARE: AN LSI-11 (M7264-YC) WITH ONE KUV11-AA, SET UP FOR ADDRESS MODE 1 OR 3 (SEE SECTION 6).

STORAGE:: KUA REQUIRES:
1. DECIMAL WORDS: 1071
2. OCTAL WORDS: 02057
3. OCTAL BYTES: 4136

3. PASS DEFINITION

SINCE THIS IS AN NBKMOD IT RUNS ONLY ONE PASS. THIS CONSISTS OF 4 ITERATIONS OF THE 'MOV1' RAM MEMORY TEST; LOADING OF THE MICRO-CODE INTO THE BOTTOM HALF OF THE KUV11-AA RAM; AND CHECKING THAT THE MICRO-CODE LOADED CORRECTLY.

4. EXECUTION TIME

KUA TAKES APPROXIMATELY 18 SECONDS, RUNNING IN AN MSV11-CD MEMORY.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DEVADR: 177540, VECTOR: 1, BR1: 0, DEVCNT: 1

REQUIRED PARAMETERS:
NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT THE KUV11-AA IS PROPERLY INSTALLED AND THAT THE MODULE (M8018) DIP SWITCHES ARE SET UP FOR EITHER ADDRESS MODE 1 OR 3.
THESE SWITCH SETTINGS ARE:

MODE	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8
1	ON	OFF	OFF	ON	OFF	ON	OFF	-
3	OFF	OFF	ON	ON	OFF	ON	OFF	-

7. MODULE OPERATION

A. RUN THE 'MOVI' MEMORY TEST ON THE KUV11-AA RAM MEMORY 4 TIMES; REPORT ERRORS ACCORDING TO SR1 BIT01 SETTING.
B. IF SR1 BIT00 = 1 AND ANY MEMORY ERROR(S) OCCURRED, THEN SKIP TO E.
C. LOAD MICRO-CODE INTO THE KUV11-AA MEMORY.
D. CHECK THAT THE MICRO-CODE WAS LOADED CORRECTLY; REPORT ERROR IF NOT.
E. EXIT THE MODULE.

8. OPERATING OPTIONS

SR1 BIT00 CLEAR(0):
LOAD THE MICRO-CODE FOR CPB AND FPA REGARDLESS OF THE RESULTS OF THE KUV11-AA RAM MEMORY TEST.

SR1 BIT00 SET(1):
IF THE RAM MEMORY TEST FAILS, DO NOT LOAD THE MICRO-CODE FOR CPB AND FPA TO RUN.

SR1 BIT01 CLEAR(0):
IN THE RAM MEMORY TEST, TYPE OUT AN ERROR MESSAGE FOR EACH ERROR ENCOUNTERED.

SR1 BIT01 SET(1):
DO NOT TYPE OUT EACH RAM MEMORY TEST ERROR; JUST TYPE ONE MESSAGE AT THE END INDICATING HOW MANY ERRORS WERE ENCOUNTERED.

9. NON-STANDARD PRINTOUTS

THE KUV11-AA MEMORY TEST ERROR REPORT DOES NOT USE THE STANDARD DEC/X11 ERROR PRINTOUT, SINCE THE MEMORY WORDS ARE 24 BITS LONG. THE ERROR PRINTOUT LOOKS AS FOLLOWS:

```
BAD DATA IN A KUV11-AA RAM WORD.
  RAM       GOOD DATA        BAD DATA
ADDRESS  CSR+4  CSR+2    CSR+4  CSR+2
```

THE FIRST 16 BITS ARE UNDER THE HEADING 'CSR+2'; THE LAST 8 BITS ARE FOUND IN THE LOWER BYTE OF 'CSR+4'. THE UPPER 8 BITS OF 'CSR+4' HAVE NO MEANING AND ARE ALWAYS ZEROES. THE 'RAM ADDRESS' CAN RANGE FROM 0 TO 1777.

```
                                        MBKMOD  <KUAB > 177540,1,0,0,1,1,163
                                                MODULE  41000,KUAB ,177540,1,0,0,1,1,163
                                                .TITLE  KUAB DEC/X11 SYSTEM EXERCISER MODULE
                                        ;       DDXCOM  VERSION  6      23-MAY-78
                                                .LIST   BIN
                                        ;*****************************************************************************
000000'                                 BEGIN:
000000' 052513  041101      040         MODNAM: .ASCII  /KUAB / ;MODULE NAME.
000005'     000                         XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177540                          ADDR:   177540+0                ;1ST DEVICE ADDR.
000010' 000001                          VECTOR: 1+0                     ;1ST DEVICE VECTOR.
000012'     000                         BR1:    .BYTE   PRTY0+0         ;1ST BR LEVEL.
000013'     000                         BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000002                          DVID1:  1+1                     ;DEVICE INDICATOR 1.
000016' 000000                          SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                          SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                          SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                          SR4:    OPEN                    ;SWITCH REGISTER 4
                                        ;*****************************************************************************
000026' 041000                          STAT:   41000                   ;STATUS WORD.
000030' 000226'                         INIT:   START                   ;MODULE START ADDR.
000032' 000224'                         SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                          PASCNT: 0                       ;PASS COUNTER.
000036' 000001                          ICONT:  1                       ;# OF ITERATIONS PER PASS=1
000040' 000000                          ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                          SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                          HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                          SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                          HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                          SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                          RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                                 CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                          RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                          RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                          SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                          SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                          SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                          SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                          SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                          SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                          SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                          CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                                 SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                          ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                                 WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                          ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                                 ERRTYP:                         ;TYPE OF ERROR
000106' 000000                          ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                          AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000226'                         RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                          WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                          WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                          INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000163                          IDNUM:  163                     ;MODULE IDENTIFICATION NUMBER=163
        000040                                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                                .NLIST
```



```
                                                .WORD   0
                                                .LIST
                                                .ENDR
000224'                                 MODSP:
                                        ;*****************************************************************************
```

```
                        ;USER EQUATES AND CONSTANTS

000001                  BIT0    = 000001
000002                  BIT1    = 000002
000004                  BIT2    = 000004
000010                  BIT3    = 000010
000020                  BIT4    = 000020
000040                  BIT5    = 000040
000100                  BIT6    = 000100
000200                  BIT7    = 000200
000400                  BIT8    = 000400
001000                  BIT9    = 001000
002000                  BIT10   = 002000
004000                  BIT11   = 004000
010000                  BIT12   = 010000
020000                  BIT13   = 020000
040000                  BIT14   = 040000
100000                  BIT15   = 100000
000000                  R0      = %0
000001                  R1      = %1
000002                  R2      = %2
000003                  R3      = %3
000004                  R4      = %4
000005                  R5      = %5
000006                  R6      = %6
000007                  R7      = %7
000006                  SP      = %6
000007                  PC      = %7

000224' 000000          TIMES:  0               ;WILL KEEP TRACK OF RAM MEMORY TEST ITERATIONS
```



```
000226'                 START:
000226' 016701 177554   RESTRT: MOV     ADDR,R1         ;GET THE BASE ADDRESS (CSR)
000232' 005011                   CLR     (R1)            ;MAKE SURE THAT ENABLE (CSR BIT12) IS CLEAR
000234' 005067 177604            CLR     HRDCNT          ;SET RAM MEMORY TEST ERROR COUNT TO ZERO
000240' 012767 000004 177756     MOV     #4,TIMES        ;SET ITERATION COUNT TO 10 OCTAL

                        ;CLEAR THE KUV11-AA RAM MEMORY

000246' 005011          CLRRAM: CLR     (R1)            ;SET UP CSR FOR Q-BUS ACCESS, RAM ADDRESS 0
000250' 012700 002000            MOV     #2000,R0        ;SET UP LOOP COUNTER
000254' 005061 000002    1$:     CLR     2(R1)           ;WRITE ZEROES...
000260' 005061 000004            CLR     4(R1)           ;...TO RAM WORD
000264' 005211                   INC     (R1)            ;INCREMENT RAM ADDRESS
000266' 005300                   DEC     R0              ;DONE CLEARING THE WCS RAM?
000270' 001371                   BNE     1$              ;BRANCH BACK AND CONTINUE IF NOT

                        ;*************************************************************************
                        ;* TEST RAM MEMORY USING 'MOVI' MEMORY TEST
                        ;*
                        ;*AFTER THE KUV11-AA RAM HAS BEEN CLEARED, THE ALGORITHM PROCEEDS AS FOLLOWS:
                        ;*
                        ;*(1) READ RAM LOCATION 0 AND VERIFY THAT IT CONTAINS ZEROES,
                        ;*    WRITE ALL ONES TO RAM LOCATION 0,
                        ;*    READ LOCATION 0 AND VERIFY THAT IT CONTAINS ALL ONES;
                        ;*(2) REPEAT THE ABOVE FOR LOCATIONS 1,2,....,1777 (I.E., ALL OF THE RAM).
                        ;*    AT THIS POINT THE RAM WILL BE FULL OF ONES.
                        ;*(3) REPEAT THE ABOVE, WRITING ZEROES THIS TIME.  AFTER THIS THE RAM WILL
                        ;*    AGAIN BE FULL OF ZEROES.
                        ;*(4) REPEAT ALL OF THE ABOVE, EXCEPT START AT THE TOP OF THE RAM AND WORK
                        ;*    DOWNWARDS (RAM LOCATIONS 1777-0).
                        ;*    AT THIS POINT THE TEST HAS SEQUENCED THROUGH THE RAM FOUR TIMES.
                        ;*(5) REPEAT THE ABOVE, EXCEPT USE BIT 1 AS THE LEAST SIGNIFICANT BIT WHEN
                        ;*    FORMING THE RAM ADDRESSES.  THE RAM ADDRESS WILL OVERFLOW AFTER CHECKING
                        ;*    ALL EVEN RAM ADDRESSES, AND THE OVERFLOW BIT IS SIMPLY WRAPPED AROUND
                        ;*    TO BIT 0 OF THE RAM ADDRESS.  TESTING CONTINUES UNTIL ALL RAM ADDRESSES
                        ;*    HAVE BEEN TESTED.
                        ;*    REPEAT AGAIN, THIS TIME WITH BIT 2 AS THE LSB IN FORMING THE RAM
                        ;*    ADDRESSES; AND AGAIN WITH BIT 3 AS THE LSB; AND SO ON UNTIL ALL BITS
                        ;*    OF THE RAM ADDRESS HAVE SERVED AS THE LSB.  SINCE THERE ARE 10 BITS
                        ;*    IN THE RAM ADDRESS (TO COVER THE RANGE 0-1777), THIS MEANS THAT STEP (5)
                        ;*    WILL BE DONE 10 TIMES IN ALL.
                        ;*************************************************************************

000272' 012702 177777   RAMTST: MOV     #177777,R2      ;R2 WILL CONTAIN TEST DATA
000276' 005003                   CLR     R3              ;R3 WILL CONTAIN TEST DATA
000300' 005004                   CLR     R4              ;R4 WILL CONTAIN THE RAM ADDRESS
000302' 012705 000001            MOV     #1,R5           ;R5 WILL HAVE THE LEAST SIGNIFICANT BIT FOR
                                                         ;GENERATING THE RAM ADDRESS
000306' 012700 002000            MOV     #2000,R0        ;SET UP LOOP COUNTER
                        ;BEGINNING OF TEST LOOP
000312' 010411          1$:      MOV     R4,(R1)         ;PUT ADDRESS INTO THE CSR
000314' 020361 000002            CMP     R3,2(R1)        ;CORRECT DATA?
000320' 001003                   BNE     2$              ;BRANCH TO ERROR IF NOT
000322' 120361 000004            CMPB    R3,4(R1)        ;CORRECT DATA?
```

```
                        BEQ     3$              ;BRANCH OVER ERROR IF OK
000326' 001404
000330' 010367  177552  2$:     MOV     R3,ASB
000334' 004767  000536          JSR     PC,MEMERR       ;GO TO ERROR REPORTING SUBROUTINE
000340' 010261  000002  3$:     MOV     R2,2(R1)        ;WRITE TEST DATA TO RAM BITS 0-15
000344' 010261  000004          MOV     R2,4(R1)        ;WRITE TEST DATA TO RAM BITS 16-23
000350' 020261  000002          CMP     R2,2(R1)        ;CORRECT DATA?
000354' 001003                  BNE     4$              ;BRANCH TO ERROR IF NOT
000356' 120261  000004          CMPB    R2,4(R1)        ;CORRECT DATA?
000362' 001404                  BEQ     5$              ;BRANCH OVER ERROR IF YES
000364' 010267  177516  4$:     MOV     R2,ASB
000370' 004767  000502          JSR     PC,MEMERR       ;GO TO ERROR REPORTING SUBROUTINE
000374' 060504  5$:     ADD     R5,R4           ;GENERATE A RAM ADDRESS
000376' 032704  002000          BIT     #BIT10,R4       ;ADDRESS OVERFLOW?
000402' 001403                  BEQ     6$              ;BRANCH IF NOT
000404' 042704  002000          BIC     #BIT10,R4       ;ADD ADDRESS OVERFLOW...
000410' 005204                  INC     R4              ;...TO BOTTOM OF ADDRESS
000412' 005300  6$:     DEC     R0              ;LOOP DONE?
000414' 001336                  BNE     1$              ;CONTINUE TEST LOOP IF NOT
000416' 005102                  COM     R2              ;R2 GETS ONES COMPLEMENT OF ITSELF
000420' 005103                  COM     R3              ;R3 GETS ONES COMPLEMENT OF ITSELF
000422' 012700  002000          MOV     #2000,R0        ;SET UP LOOP COUNTER
000426' 005004                  CLR     R4              ;INITIALIZE THE RAM ADDRESS
000430' 005702                  TST     R2              ;HAVE WE GONE THROUGH TWICE?
000432' 001727                  BEQ     1$              ;NO, GO BACK AND DO IT AGAIN WITH COMPLEMENTED DATA
                        ;DO THE SAME AS ABOVE EXCEPT TEST RAM MEMORY FROM TOP TO BOTTOM (1777-0)
000434' 012704  001777          MOV     #1777,R4        ;SET UP RAM ADDRESS
000440' 010411  7$:     MOV     R4,(R1)         ;PUT ADDRESS INTO THE CSR
000442' 020361  000002          CMP     R3,2(R1)        ;CORRECT DATA?
000446' 001003                  BNE     8$              ;BRANCH TO ERROR IF NOT
000450' 120361  000004          CMPB    R3,4(R1)        ;CORRECT DATA?
000454' 001404                  BEQ     9$              ;BRANCH OVER ERROR IF OK
000456' 010367  177424  8$:     MOV     R3,ASB
000462' 004767  000410          JSR     PC,MEMERR       ;GO TO MEMORY ERROR REPORTING SUBROUTINE
000466' 010261  000002  9$:     MOV     R2,2(R1)        ;WRITE TEST DATA TO RAM BITS 0-15
000472' 010261  000004          MOV     R2,4(R1)        ;WRITE TEST DATA TO RAM BITS 16-23
000476' 020261  000002          CMP     R2,2(R1)        ;CORRECT DATA?
000502' 001003                  BNE     10$             ;BRANCH TO ERROR IF NOT
000504' 120261  000004          CMPB    R2,4(R1)        ;CORRECT DATA?
000510' 001404                  BEQ     11$             ;BRANCH OVER ERROR IF YES
000512' 010267  177370  10$:    MOV     R2,ASB
000516' 004767  000354          JSR     PC,MEMERR       ;GO TO MEMORY ERROR REPORTING SUBROUTINE
000522' 160504  11$:    SUB     R5,R4           ;GENERATE A RAM ADDRESS
000524' 100002                  BPL     12$             ;BRANCH IF NO ADDRESS UNDERFLOW OCCURED
000526' 062704  001777          ADD     #1777,R4        ;CORRECT THE ADDRESS UNDERFLOW
000532' 005300  12$:    DEC     R0              ;LOOP DONE?
000534' 001341                  BNE     7$              ;CONTINUE TEST LOOP IF NOT
000536' 005102                  COM     R2              ;R2 GETS ONES COMPLEMENT OF ITSELF
000540' 005103                  COM     R3              ;R3 GETS ONES COMPLEMENT OF ITSELF
000542' 012700  002000          MOV     #2000,R0        ;SET UP LOOP COUNTER
000546' 012704  001777          MOV     #1777,R4        ;INITIALIZE THE RAM ADDRESS
000552' 005702                  TST     R2              ;HAVE WE GONE THROUGH TWICE?
000554' 001731                  BEQ     7$              ;GO BACK AND DO IT AGAIN WITH COMPLEMENTED DATA IF NOT

                        ;GENERATE A NEW LEAST SIGNIFICANT BIT FOR RAM ADDRESS GENERATION AND,
                        ;IF NOT DONE WITH THE TEST, GO BACK AND RUN SOME MORE
000556' 006305                  ASL     R5              ;SHIFT THE LSB FOR SUBSEQUENT RAM ADDRESS GENERATION
```



```
000560' 005004                  CLR     R4              ;INITIALIZE THE RAM ADDRESS
000562' 032705  002000          BIT     #BIT10,R5       ;SEE IF WE ARE DONE WITH THE RAM MEMORY TEST
000566' 001002                  BNE     13$             ;BRANCH IF YES
000570' 000167  177516          JMP     1$              ;CONTINUE TESTING IF NOT
000574' 005367  177424  13$:    DEC     TIMES           ;ALL ITERATIONS DONE?
000600' 001402                  BEQ     14$             ;BRANCH IF YES
000602' 000167  177440          JMP     CLRRAM          ;NO, RUN RAM MEMORY TEST AGAIN

000606' 005767  177232  14$:    TST     HRDCNT          ;ANY RAM MEMORY ERRORS?
000612' 001410                  BEQ     15$             ;BRANCH IF NOT
                                ;*****************************************************************
                                ;CONVERT HRDCNT TO ASCII AND
                                ;STORE AT COUNT
000614' 104420  000000' 000044'         OTOA$,BEGIN,HRDCNT,COUNT
000622' 001614'
                                ;*****************************************************************
000624' 104403  000000' 001242'         MSGN$,BEGIN,ERR2        ;ASCII MESSAGE CALL WITH COMMON HEADER
000632' 000403                  BR      16$             ;SKIP PRINTOUT
000634'                 15$:
000634' 104403  000000' 001246'         MSGN$,BEGIN,RAMOK       ;ASCII MESSAGE CALL WITH COMMON HEADER

                        ;*****************************************************************************
                        ;*
                        ;* THE NEXT ROUTINE LOADS MICRO-CODE FROM THE TABLES AT 'UCODE' & 'UCODE1'  INTO
                        ;* THE KUV11-AA RAM AND CHECKS THAT IT LOADED CORRECTLY.
                        ;*
                        ;*****************************************************************************

                        ;LOAD THE MICRO-CODE
000642' 036727  177150  000001  16$:    BIT     SR1,#BIT0       ;LOAD MICRO-CODE EVEN IF RAM MEMORY TEST HAD ERRORS?
000650' 001403                  BEQ     17$             ;BRANCH IF YES
000652' 005767  177166          TST     HRDCNT          ;SEE IF ANY RAM MEMORY TEST ERRORS OCCURED
000656' 001103                  BNE     ABORT           ;YES, BRANCH TO LOADER ABORT
000660' 005011  17$:    CLR     (R1)            ;SET RAM ADDRESS TO 0
000662' 012702  002070'         MOV     #UCODE,R2       ;GET FIRST MICRO-CODE TABLE ADDRESS
000666' 012261  000002  18$:    MOV     (R2)+,2(R1)     ;LOAD A KUV11-AA RAM WORD, BITS 15-0
000672' 021127  000777          CMP     (R1),#777       ;DONE?
000676' 001402                  BEQ     19$             ;BRANCH IF YES
000700' 005211                  INC     (R1)            ;INCREMENT THE RAM ADDRESS
000702' 000771                  BR      18$             ;BRANCH BACK AND CONTINUE LOADING MICRO-CODE
000704' 012702  004070'         MOV     #UCODE1,R2      ;GET SECOND MICRO-CODE TABLE ADDRESS
000710' 012200  20$:    MOV     (R2)+,R0        ;GET A TABLE ENTRY
000712' 001414                  BEQ     22$             ;BRANCH OUT OF MICRO-CODE LOADER IF A ZERO TABLE
                                                        ;ENTRY IS ENCOUNTERED
000714' 012703  000001          MOV     #BIT0,R3        ;PRE-LOAD R3 WITH RAM DATA
000720' 005700                  TST     R0              ;SEE IF BIT15 IS SET
000722' 100004                  BPL     21$             ;BRANCH IF NOT
000724' 012703  000002          MOV     #BIT1,R3        ;PRE-LOAD R3 WITH RAM DATA
000730' 042700  100000          BIC     #BIT15,R0       ;STRIP OFF BIT 15
000734' 010011  21$:    MOV     R0,(R1)         ;LOAD THE RAM ADDRESS
000736' 010361  000004          MOV     R3,4(R1)        ;LOAD THE RAM WORD, BITS 23-16
000742' 0007b2                  BR      20$
                        ;CHECK FOR CORRECT MICRO-CODE LOAD
000744' 005011  22$:    CLR     (R1)            ;SET RAM ADDRESS TO 0
000746' 012702  002070'         MOV     #UCODE,R2       ;GET FIRST MICRO-CODE TABLE ADDRESS
000752' 022261  000002  23$:    CMP     (R2)+,2(R1)     ;CORRECT DATA IN RAM, BITS 15-0?
```

```
000756' 001026                          BNE     27$             ;BRANCH TO ERROR IF NOT
000760' 021127  000777                  CMP     (R1),#777       ;DONE?
000764' 001402                          BEQ     24$             ;BRANCH IF YES
000766' 005211                          INC     (R1)            ;INCREMENT RAM ADDRESS
000770' 000770                          BR      23$             ;BRANCH BACK AND CONTINUE CHECKING MICRO-CODE
000772' 012702  004070'         24$:    MOV     #UCODE1,R2      ;GET SECOND MICRO-CODE TABLE ADDRESS
000776' 012200                  25$:    MOV     (R2)+,R0        ;GET A TABLE ENTRY
001000' 001423                          BEQ     26$             ;BRANCH OUT OF MICRO-CODE CHECKER IF A ZERO TABLE
                                                                ;ENTRY IS ENCOUNTERED
001002' 012703  000001                  MOV     #BIT0,R3        ;PRE-LOAD R3 WITH EXPECTED RAM DATA
001006' 005700                          TST     R0              ;SEE IF BIT15 IS SET
001010' 100004                          BPL     26$             ;BRANCH IF NOT
001012' 012703  000002                  MOV     #BIT1,R3        ;PRE-LOAD R3 WITH EXPECTED RAM DATA
001016' 042700  100000                  BIC     #BIT15,R0       ;STRIP OFF BIT 15
001022' 010011                  26$:    MOV     R0,(R1) ;LOAD THE RAM ADDRESS
001024' 020361  000004                  CMP     R3,4(R1)        ;CHECK FOR CORRECT DATA IN RAM BITS 23-16
001030' 001001                          BNE     27$             ;BRANCH TO ERROR IF BAD
001032' 000761                          BR      25$             ;CONTINUE CHECKING

001034'                         27$:
001034' 104403  000000' 001256'         MSGN$,BEGIN,LDBAD       ;ASCII MESSAGE CALL WITH COMMON HEADER
001042' 005267  176776                  INC     #RDCNT          ;ADD THIS LOAD ERROR TO THE ERROR COUNT
001046' 000405                          BR      END

001050'                         28$:
001050' 104403  000000' 001252'         MSGN$,BEGIN,LOADOK      ;ASCII MESSAGE CALL WITH COMMON HEADER
001056' 052711  010000                  BIS     #BIT12,(R1)     ;SET KUV11-AA ENABLE BIT, TO ALLOW MIB ACCESS TO THE RAM
001062'                         END:
001062' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS

001066'                         ABORT:
001066' 104403  000000' 001262'         MSGN$,BEGIN,LDABRT      ;ASCII MESSAGE CALL WITH COMMON HEADER
001074' 000772                          BR      END


                                ;*************************************************************************
                                ;*
                                ;*SUBROUTINE    MEMERR
                                ;*
                                ;*THIS SUBROUTINE REPORTS KUV11-AA RAM MEMORY ERRORS.  EXPECTED DATA IS PASSED
                                ;*VIA 'ASB'; THE BAD DATA AND FAILING RAM ADDRESS ARE OBTAINED THROUGH THE KUV11-AA
                                ;*DEVICE REGISTERS.  THE BASE REGISTER ADDRESS OF THE KUV11-AA IS PASSED VIA
                                ;*'ADDR'.  THE STANDARD MONITOR CALLS 'MSGN' AND 'DACNV' ARE MADE IN THE SUBTOUTINE,
                                ;*AND USE GENERAL REGISTER 5.  NO OTHER GENERAL PURPOSE REGISTERS ARE USED.
                                ;*THIS SUBROUTINE IS CALLED AS FOLLOWS:          JSR     PC,MEMERR
                                ;*
                                ;*************************************************************************
001076' 005267  176742          MEMERR: INC     #RDCNT          ;COUNT THE NUMBER OF RAM MEMORY ERRORS
001102' 036727  176710  000002          BIT     SR1,#BIT1       ;REPORT EACH ERROR?
001110' 001050                          BNE     1S              ;BRANCH OVER ERROR REPORTING IF NO
                                        ;*****************************************************************
                                                                ;CONVERT ASB TO ASCII AND
                                                                ;STORE AT GOODLO
001112' 104420  000000' 000106'         OTOA$,BEGIN,ASB,GOODLO
001120' 001462'
```



```
                                        ;*****************************************************************
001122' 042767  177400  176756           BIC     #177400,ASB     ;STRIP OFF HIGH BYTE
                                        ;*****************************************************************
                                                                ;CONVERT ASB TO ASCII AND
                                                                ;STORE AT GOODHI
001130' 104420  000000' 000106'         OTOA$,BEGIN,ASB,GOODHI
001136' 001453'

                                        ;*****************************************************************
001140' 011167  000070                  MOV     (R1),GUNCH      ;CONVERT FAILING RAM ADDRESS TO ASCII
                                        ;*****************************************************************
                                                                ;CONVERT GUNCH TO ASCII AND
                                                                ;STORE AT ADDRES
001144' 104420  000000' 001234'         OTOA$,BEGIN,GUNCH,ADDRES
001152' 001442'

                                        ;*****************************************************************
001154' 112767  000040  000260          MOVB    #40,ADDRES      ;PUT TWO ASCII SPACES INTO ADDRES
001162' 112767  000040  000253          MOVB    #40,ADDRES+1    ;
001170' 016167  000002  000036          MOV     2(R1),GUNCH     ;CONVERT BAD DATA TO ASCII
                                        ;*****************************************************************
                                                                ;CONVERT GUNCH TO ASCII AND
                                                                ;STORE AT BADLO
001176' 104420  000000' 001234'         OTOA$,BEGIN,GUNCH,BADLO
001204' 001502'

                                        ;*****************************************************************
001206' 016167  000004  000020          MOV     4(R1),GUNCH     ;CONVERT BAD DATA TO ASCII
                                        ;*****************************************************************
                                                                ;CONVERT GUNCH TO ASCII AND
                                                                ;STORE AT BADHI
001214' 104420  000000' 001234'         OTOA$,BEGIN,GUNCH,BADHI
001222' 001473'

                                        ;*****************************************************************
                                                                ;TYPE OUT THE ERROR MESSAGE AND DATA
001224' 104403  000000' 001236'         MSGN$,BEGIN,ERR1        ;ASCII MESSAGE CALL WITH COMMON HEADER
001232' 000207                  1S:     RTS     PC              ;RETURN FROM THE SUBROUTINE
```

```
001234' 000000                          GUNCH:  .WORD   0                       ;TEMP LOC. FOR OCTAL TO ASCII CONV.

001236' 001266'                         ERR1:   ERROR1
001240' 177777                                  177777

001242' 001511'                         ERR2:   ERROR2
001244' 177777                                  177777

001246' 001625'                         RAMOK:  MEMOK
001250' 177777                                  177777

001252' 001667'                         LOADOK: LOK
001254' 177777                                  177777

001256' 001731'                         LDBAD:  LBAD
001260' 177777                                  177777

001262' 001775'                         LDABRT: LABORT
001264' 177777                                  177777


001266' 041045  042101  042040  ERROR1: .ASCII  '%BAD DATA IN A KUV11-AA RAM WORD.'
001274' 052101  020101  047111
001302' 040440  045440  053125
001310' 030461  040455  020101
001316' 040522  020115  047527
001324' 042122     056
001327'    045  020040  040522          .ASCII  '%  RAM      GOOD DATA       BAD DATA'
001334' 020115  020040  020040
001342' 043440  047517  020104
001350' 040504  040524  020040
001356' 020040  020040  041040
001364' 042101  042040  052101
001372'    101
001373'    045  042101  051104          .ASCII  '%ADDRESS  CSR+4  CSR+2    CSR+4  CSR+2%'
001400' 051505  020123  041440
001406' 051123  032053  020040
001414' 051503  025522  020062
001422' 020040  041440  051123
001430' 032053  020040  051503
001436' 025522  022462

001442' 000006                          ADDRES: .BLKB   6               ;BAD RAM ADDRESS
001450'    040     040     040                  .BYTE   40,40,40        ;ASCII SPACES
001453' 000006                          GOODHI: .BLKB   6               ;EXPECTED DATA, BITS 23-16
001461'    040                                  .BYTE   40              ;ASCII SPACE
001462' 000006                          GOODLO: .BLKB   6               ;EXPECTED DATA, BITS 15-0
001470'    040     040     040                  .BYTE   40,40,40        ;ASCII SPACES
001473' 000006                          BADHI:  .BLKB   6               ;ACTUAL DATA, BITS 23-16
001501'    040                                  .BYTE   40              ;ASCII SPACE
001502' 000006                          BADLO:  .BLKB   6               ;ACTUAL DATA, BITS 15-0
001510'    000                                  .BYTE   0               ;ASCII MESSAGE TERMINATOR
```



```
001511'    045  040522  020115  ERROR2: .ASCII  '%RAM MEMORY TEST UNSUCCESSFUL; TOTAL NUMBER OF ERRORS (IN OCTAL) = '
001516' 042515  047515  054522
001524' 052040  051505  020124
001532' 047125  052523  041503
001540' 051505  043123  046125
001546' 020073  047524  040524
001554' 020114  052516  041115
001562' 051105  047440  020106
001570' 051105  047522  051522
001576' 024040  047111  047440
001604' 052103  046101  020051
001612' 020075
001614' 000006                          COUNT:  .BLKB   6
001622'    015                                  .BYTE   15
001623'    012                                  .BYTE   12
001624'    000                                  .BYTE   0               ;ASCII MESSAGE TERMINATOR

001625'    045  040522  020115  MEMOK:  .ASCIZ  '%RAM MEMORY TEST WAS SUCCESSFUL.%'
001632' 042515  047515  054522
001640' 052040  051505  020124
001646' 040527  020123  052523
001654' 041503  051505  043123
001662' 046125  022456     000


001667'    045  044515  051103  LOK:    .ASCIZ  '%MICRO-CODE LOAD WAS SUCCESSFUL.%'
001674' 026517  047503  042504
001702' 046040  040517  020104
001710' 040527  020123  052523
001716' 041503  051505  043123
001724' 046125  022456     000

001731'    045  051105  047522  LBAD:   .ASCIZ  '%ERROR OCCURED IN MICRO-CODE LOAD.%'
001736' 020122  041517  052503
001744' 042522  020104  047111
001752' 046440  041511  047522
001760' 041455  042117  020105
001766' 047514  042101  022456
001774'    000
001775'    045  044515  051103  LABORT: .ASCIZ  '%MICRO-CODE LOAD IS ABORTED SINCE RAM MEMORY TEST FAILED.%'
002002' 026517  047503  042504
002010' 046040  040517  020104
002016' 051511  040440  047502
002024' 052122  042105  051440
002032' 047111  042503  051040
002040' 046501  046440  046505
002046' 051117  020131  042524
002054' 052123  043040  044501
002062' 042514  027104  000045

                                        .EVEN
```

```
  633
  634
  635  002070' 072411                UCODE:  072411
  636                                        ;*
  637                                        ;*
  638
  639
  640  004070' 000123                UCODE1: 000123
  641                                        ;*
  642                                        ;*
  643  004134' 000000                        000000
  644
  645
  646          000001                .END
```



```
ABORT   001066R          406     461#
ACSR    000102R          210#
ADDR    000006R          176#    265
ADDRES  001442R          498     501#    502*    567#
ADDR22= 001000           228#
ASB     000106R          214#    319#    327*    349*    357*    484     487*    491
ASTAT   000104R          212#
AWAS    000110R          215#
BADHI   001473R          514     573#
BADLO   001502R          507     575#
BEGIN   000000R          173#    387     390     393     450     455     458     462     484     491     498     507     514
                         518
BIT0  = 000001           228#    232#    403     418     439
BIT1  = 000002           228#    233#    421     442     479
BIT10 = 002000           228#    242#    330     332     375
BIT11 = 004000           228#    243#
BIT12 = 010000           228#    244#    456
BIT13 = 020000           228#    245#
BIT14 = 040000           228#    246#
BIT15 = 100000           228#    247#    422     443
BIT2  = 000004           228#    234#
BIT3  = 000010           228#    235#
BIT4  = 000020           228#    236#
BIT5  = 000040           228#    237#
BIT6  = 000100           228#    238#
BIT7  = 000200           228#    239#
BIT8  = 000400           228#    240#
BIT9  = 001000           228#    241#
BREAKS= 104407           228#
BR1     000012R          178#
BR2     000013R          179#
BTOD$ = 104421           228#
CDATA$= 104412           228#
CLRRAM  000246R          272#    380
CONFIG  000056R          198#
COUNT   001614R          387     593#
CSRA    000100R          208#
DATCK$= 104411           228#
DATER$= 104404           228#
DVID1   000014R          180#
END     001062R          452     457#    463
ENDIT$= 104413           228#    458
END$  = 104410           228#
ERROR1  001266R          527     546#
ERROR2  001511R          530     581#
ERRTYP  000106R          213#
ERR1    001236R          518     527#
ERR2    001242R          390     530#
EXITS = 104400           228#
GETPA$= 104415           228#
GOODHI  001453R          491     569#
GOODLO  001462R          484     571#
GUNCH   001234R          494*    498     503*    507     510*    514     525#
GWBUF$= 104414           228#
HRDCNT  000044R          193#    267#    382     387     405     451*    478*
HRDERS= 104405           228#
```

```
HRDPAS  000050R          195#
ICONT   000036R          190#
ICOUNT  000040R          191#
IDNUM   000122R          220#
INIT    000030R          187#
INTR    000120R          219#
LABORT  001775R          542     620#
LBAD    001731R          539     613#
LDABRT  001262R          462     542#
LDBAD   001256R          450     539#
LOADOK  001252R          455     536#
LOK     001667R          536     606#
MAP22$= 104416           228#
MEMERR  001076R          320     328     350     358     478#
MEMOK   001625R          533     598#
MODNAM  000000R          174#
MODSP   000224R          188     226#
MSGN$ = 104403           228#    390     393     450     455     462     518
MSGS$ = 104402           228#
MSG$  = 104401           228#
NULL  = 000000           228#
OPEN  = 000000           175     181     182     183     184     201     202     203     204     205     206     207     208
                         210     212     214     215     217     218     219     228#
OTOA$ = 104420           228#    387     484     491     498     507     514
PASCNT  000034R          189#
PIRQ$ = 000004           228#
POPSP = 005726           228#
POPSP2= 022626           228#
PRIY  = 000000           228#
PRIY0 = 000000           178     179     228#
PRIY1 = 000040           228#
PRIY2 = 000100           228#
PRIY3 = 000140           228#
PRIY4 = 000200           228#
PRIY5 = 000240           228#
PRIY6 = 000300           228#
PRIY7 = 000340           228#
PS    = 177776           228#
PSW   = 177776           228#
PUSH  = 005746           228#
PUSH2 = 024646           228#
RAMOK   001246R          393     533#
RAMTST  000272R          307#
RAND$ = 104417           228#
RANNUM  000054R          197#
RESTRT  000226R          216     265#
RES1    000056R          199#
RES2    000060R          200#
RSTRT   000112R          216#
SBADR   000102R          209#
SUFCNT  000042R          192#
SOFER$= 104406           228#
SOFPAS  000046R          194#
SPOINT  000032R          188#
SPSIZ = 000040             1#    221
SR1     000016R          181#    403     479
```



```
SR2     000020R          182#
SR3     000022R          183#
SR4     000024R          184#
START   000226R          187     264#
STAT    000026R          186#
SVR0    000062R          201#
SVR1    000064R          202#
SVR2    000066R          203#
SVR3    000070R          204#
SVR4    000072R          205#
SVR5    000074R          206#
SVR6    000076R          207#
SYSCNT  000052R          196#
TIMES   000224R          260#    268*    378*
TRPDFD= 000022           228#
UCODE   002070R          408     428     635#
UCODE1  004070R          414     435     640#
VECTOR  000010R          177#
WASADR  000104R          211#
WDFR    000116R          218#
WDTO    000114R          217#
XFLAG   000005R          175#
.     = 004136R          567#    569#    571#    573#    575#    593#

. ABS.  000000     000
        004136     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XKUAB0,XKUAB0/SOL/CRF:SYM=DDXCOM,XKUAB0
 RUN-TIME:  1  2  .2 SECONDS
 RUN-TIME RATIO: 36/4=7.8
 CORE USED:  7K  (13 PAGES)
```

```
1          .NLIST BIN
2          .REM   %
3
4
5                                   IDENTIFICATION
6                                   --------------
7
8                  PRODUCT CODE:        AC-E974C-MC
9
10                 PRODUCT NAME:        CXLPHC0 LPA11-XX MODULE
11
12                 PRODUCT DATE:        APRIL 1981
13
14                 MAINTAINER:          DEC/X11 SUPPORT GROUP
15
16                 THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE
17                 WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT
18                 BY DIGITAL EQUIPMENT CORPORATION.  DIGITAL EQUIPMENT
19                 CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT
20                 MAY APPEAR IN THIS MANUAL.
21
22                 THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE
23                 PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER
24                 SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITALS
25                 COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY
26                 OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.
27
28                 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR
29                 THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS
30                 NOT SUPPLIED BY DIGITAL.
31
32               COPYRIGHT (C) 1978, 1980, 1982    DIGITAL EQUIPMENT CORPORATION
```



```
33
34          1.0    ABSTRACT
35                 --------
36
37                 LPH IS AN IOMOD THAT EXERCISES THE LPA11-XX OPTION. NO DATA
38                 WRAPAROUND THRU THE DIGITAL I-O OR ANALOG CHANNELS ARE CHECKED.
39                 WHEN MULTI-USER MICRO-CODE (DEFAULT) IS LOADED IN THE LPA-11,
40                 ALL SELECTED DEVICES ON THE LPA-11'S I/O BUS ARE EXERCISED.
41                 WHEN DEDICATED USER MICRO-CODE IS LOADED (SR1 BIT 6 SET) IN
42                 THE LPA11, ONLY A/D'S ON THE LPA11'S I/O BUS ARE EXERCISED
43                 AND HIGH-SPEED SAMPLE TAKING IS ENABLED. A 150 KILO-HERTZ SAMPLE
44                 RATE IS THE DEFAULT USED. SAMPLING AT THIS RATE MAY PRODUCE DATA
45                 LATENCY PROBLEMS IN OTHER NPR DEVICES ON THE PDP-11 SYSTEM BUS.
46                 IN ADDITION, OTHER DEVICES MAY CAUSE LOSS OF PROGRAM EXECUTION
47                 SPEED AS TO CAUSE AN LPA-11 "USER REQUEST" ERROR CODE 241 (HIGH-
48                 BYTE OF "CSRC" ERROR TYPEOUT). FOR EXAMPLE, WHEN RUNNING AN RK11
49                 AND DEDICATED LPA-11 MICRO-CODE WITH ONE A TO D, SR2 SHOULD BE
50                 SET TO A 1. WHEN RUNNING WITH TWO A TO D'S, SR2 SHOULD BE SET TO A 5.
51                 THE VALUE OF "SR2" DETERMINE THE SAMPLE RATE USED FOR THE A/D.
52
53          2.0    REQUIREMENTS
54                 ------------
55
56                 HARDWARE:  ONE LPA-11XX INCLUDES:
57                        1 KMC-11, 1 M8200-YC, 1 M8254, 1 CLOCK (KW11K, LPSKW OR AR11)
58                        PLUS AT LEAST ONE OF THE FOLLOWING DEVICES:
59                               AD11K, AA11K, DR11K, AR11, LPSDR, LPSAD, LPSVC
60
61                 STORAGE: LPH REQUIRES:
62                               1. DECIMAL WORDS:        5494
63                               2. OCTAL WORDS:         12566
64                               3. OCTAL BYTES:         25354
65
66          3.0    PASS DEFINITION
67                 ---------------
68
69                 THE NUMBER OF NPR DATA TRANSFERS FOR A SUCCESSFULL PASS IS
70                 COMPLETELY DEPENDANT UPON THE LPA-11 CONFIGURATION.
71
72          4.0    EXECUTION TIME
73                 --------------
74
75                 ONE PASS OF LPH WITH MULTI USER (DEFAULT 10KHZ):
76                        1 CLOCK AND 1 A TO D =          60 SECONDS
77                        1 CLOCK AND 2 A TO D =          60 SECONDS
78                        1 CLOCK AND 1 DIGITAL I/O =     90 SECONDS
79                        1 CLOCK AND 1 D TO A =          90 SECONDS
80                 ONE PASS OF LPH WITH DEDICATED A/D MICRO-CODE (150KHZ):
81                        1 CLOCK AND 1 A TO D =          10 SECONDS
82                        1 CLOCK AND 2 A TO D =          10 SECONDS
83                 ONE PASS OF LPH WITH DEDICATED A/D MICRO-CODE (20KHZ):
84                        1 CLOCK AND 1 A TO D =          20 SECONDS
85                        1 CLOCK AND 2 A TO D =          20 SECONDS
86
87
```

```
                5.0     CONFIGURATION REQUIREMENTS
                        --------------------------

                        DEFAULT PARAMETERS:

                        DEVADR:  170460, VECTOR:  300, BR1: 5

                        DEVCNT:  N/A, SR1:  0, SR2:  0

                        REFER TO THE PROGRAM LISTING SECTION FOR THE ADDRESS
                        ASSIGNMENTS OF THE DEVICES ON THE LPA-11 I/O BUS.

                        REQUIRED PARAMETERS

                        SR1     MUST BE SET UP TO REFLECT THE TRUE LPA11-KX
                                SYSTEM CONFIGURATION (SEE OPERATIONS OPTIONS)
                        NOTE: THIS MODULE WILL ERROR AND WILL BE "DROPPED"
                                UNLESS A CLOCK HAS BEEN SPECIFIED IN SR1:
                        SR2     MAY BE CHANGED TO LOWER THE A/D TRANSFER RATE
                                IF DEDICATED MICRO-CODE HAS BEEN SELECTED.

                6.0     DEVICE/OUTPUT SET-UP
                        --------------------

                        ALL CONNECTIONS TO THE A TO D, D TO A AND CLOCK INPUT
                        OR OUTPUT CONNECTOR SHOULD BE REMOVED.

                7.0     MODULE OPERATION
                        ----------------

                        1.  "TST1" LOADS AND VERIFIES EITHER DEDICATED OR MULTI-USER
                            MICRO-CODE INTO THE KMC-11 AS INDICATED BY BIT 6 OF "SR1".

                        2.  "TST2" CHECKS THAT THE MICRO-CODE CAN BE STARTED.

                        3.  "TST3" BUILDS A DEVICE ADDRESS TABLE BASED ON THE INFORMATION
                            THE OPERATOR FURNISHED IN SR1.  AFTER BUILDING THE
                            TABLE, IT GIVES THE TD THE LPA-11 FOR VERIFICATION.

                        4.  "TST4" DETERMINES THE LOCATION OF THE LPA-11K BUFFERS IN
                            18 BIT ADDRESSING. THE LIST OF "START DATA TRANSFER" IS
                            CREATED AS "TO DO" QUEUE JOB ENTRIES. THE LPA-11 CLOCK IS
                            STARTED AND THE INITIATION OF THE "TO DO" QUEUE JOBS OCCURS.
                            WHEN MULTI-USER MICRO-CODE IS SELECTED <DEFAULT>, ALL
                            DEVICES ENABLED, VIA SR1, ON THE LPA-11 I/O BUS ARE EXERCISED.
                            IF DEDICATED CODE IS SELECTED, A/D SAMPLES ARE TAKEN AT A RATE
                            CONTROLLED BY THE VALUE IN SR2.
```



```
                8.0     OPERATIONS OPTIONS
                        ---------- -------

                        SR1: INFORMS THE LPH MODULE AS TO WHAT DEVICES ARE ON THE
                        LPA-11 I/O  BUS AND THE MODE THE LPA-11 WILL BE RUNNING.
                        YOU MUST CHANGE THIS LOCATION TO REFLECT THE ACTUAL
                        SYSTEM CONFIGURATION.

                        SR 1 BIT=1      OCTAL   DEVICE
                        ----------      -----   ------
                            0           000001  1ST AD11K
                            1           000002  1ST DA11K
                            2           000004  1ST DR11K
                            3           000010  1ST AA11K
                            4           000020  2ND AD11K
                            5           000040  2ND DR11K
                            6           000100  DEDICATED USER MICRO-CODE SELECTED
                            7           000200  3RD DR11K
                            8           000400  4TH DR11K
                            9           001000  5TH DR11K
                           10           002000  AR11K
                           11           004000  UNUSED
                           12           010000  LPSAD (LPS A/D)
                           13           020000  LPSKW (LPS REAL TIME CLOCK)
                           14           040000  LPSVC (LPS D/A)
                           15           100000  LPSDR (LPS DIGITAL I/O)

                        NOTE:   BEFORE RUNNING THIS MODULE, ALL CONNECTIONS TO THE
                                A TO D, D TO A AND CLOCK INPUT OR OUTPUT CONNECTOR
                                SHOULD BE REMOVED. IF NOT, IT IS UNPREDICTABLE WHAT
                                THESE DEVICE WILL DO.

                        SR2: INFORMS THE MODULE AS TO THE A/D DATA TRANSFER RATE

                                OCTAL VALUE             RATE (KILO-HERTZ)
                                -----------             ----

                                    0                   150K
                                    1                   134K
                                    2                   118K
                                    3                   100K
                                    4                    80K
                                    5                    60K
                                    6                    40K
                                    7                    20K

                9.0     NON-STANDARD PRINTOUTS
                        ----------------------

                        ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED
                        IN THE DEC/X11 DOCUMENT.
```

```
                .LIST BIN
                %
000000'         IOMOD   <LPHC >,170460,300,5,,0,50,0
000000'         MODULE  140000,LPHC ,170460,300,5,0,0,50,0
                .TITLE  LPHC DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                        ;*****************************************************************************
000000'                 BEGIN:
000000' 050114  041510      040 MODNAM: .ASCII  /LPHC / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 170460                  ADDR:   170460+0                ;1ST DEVICE ADDR.
000010' 000300                  VECTOR: 300+0                   ;1ST DEVICE VECTOR.
000012'     240                 BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                        ;*****************************************************************************
000026' 140000                  STAT:   140000                  ;STATUS WORD.
000030' 000506'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000050                  ICONT:  50                      ;# OF ITERATIONS PER PASS=50
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SHADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASP:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000506'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTU:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000000                  IDNUM:  0                       ;MODULE IDENTIFICATION NUMBER=0
```



```
000224'                 MODSP:
                        ;*****************************************************************************
000224' 000001          MODREV: 1                                       ;NEW LPHC   PA001
                
000226' 177763          SR2VAL: -13.                    ;CLOCK PRESET VALUE FOR SR2 RATE CODE
000230'                 KMCSR:
                                ;
                                ;ADDRESS OF KMC-11 OF LPA-11    THE ADDR FOR KMAD0 MAY BE
                                ;                               CHANGED BY THE USER TO REFLECT
                                ;                               A DIFFERENT KMC-11 ADDR. THE
                                ;                               REST OF THE ADDRESSES WILL
                                ;                               BE CHANGED BY THE PROGRAM.
                                ;
000230'                 LPCI:
000230' 170460          KMAD0:  .WORD   170460          ;BASE KMC ADDR. MAY BE PATCHED BY USER.
000232'                 LPMR:
000232' 170461          KMAD1:  .WORD   170460+1                ;>DO NOT        <;KMC-CSR ADDR
000234'                 LPC0:
000234' 170462          KMAD2:  .WORD   170460+2                ;>PATCH         <;
000236'                 LPSG:
000236' 170463          KMAD3:  .WORD   170460+3                ;>THIS AREA     <
000240'                 LPALL:
000240' 170464          KMAD4:  .WORD   170460+4                ;
000242'                 LPADM:
000242' 170465          KMAD5:  .WORD   170460+5                ;>DO NOT        <
000244'                 LPMS1:
000244' 170466          KMAD6:  .WORD   170460+6                ;>PATCH         <
000246'                 LPMS2:
000246' 170467          KMAD7:  .WORD   170460+7                ;>THIS AREA     <

000250' 000005          VERSN:  .WORD   5               ;CURRENT VERSION NUMBER OF MICROCODE.

000252' 170400          AD11K:  .WORD   170400          ;AD11K #1 ADDRESS.
000254' 170404          KW11K:  .WORD   170404          ;KW11K ADDRESS.
000256' 167770          DR11K1: .WORD   167770          ;DR11K #1 ADDR.
000260' 170416          AA11K:  .WORD   170416          ;AA11 ADDRESS.
000262' 170440          AD11K2: .WORD   170440          ;AD11K #2 ADDRESS.
000264' 167760          DR11K2: .WORD   167760          ;DR11K #2 ADDRESS.
000266' 167750          DR11K3: .WORD   167750          ;DR11K #3 ADDRESS.
000270' 167740          DR11K4: .WORD   167740          ;DR11K #4 ADDRESS.
000272' 167730          DR11K5: .WORD   167730          ;DR11K #5 ADDRESS.
000274' 170400          AR11K:  .WORD   170400          ;AR11K ADDRESS.
000276' 170400          LPS11:  .WORD   170400          ;LPS11 BASE ADDRESS.
```

```
                                ;
000300' 000115          UCODE:  .ASCIZ  /M/             ;TYPE OF U CODE SELECTED
000302' 015346'         UCODES: .WORD   MMAST            ;POINTER TO SELECTED U CODE
000304' 000000          $BDDAT: .WORD   0               ;TEMP STORAGE.
000306' 000000          $GDDAT: .WORD   0               ;TEMP   STORAGE.
                                ;
000310' 007326'         GETADR: .WORD   BUFFER          ;THIS LOCATION USED TO GET PA BITS.
000312' 000000          GETPAB: .WORD   0               ;CONTAINS PA BITS (FROM MONITOR)
000314' 000000          GETEAB: .WORD   0               ;CONTAINS EA BITS SHIFTED.
000316' 007326'         MBUFF0: .WORD   BUFF0           ;THIS LOC USED TO GET THE PA OF THE BUFFER
000320' 000000          GBUFF0: .WORD   0               ;CONTAINS PA BITS (FROM MONITOR)
000322' 000000          EBUFF0: .WORD   0               ;CONTAINS EA BITS SHIFTED
000324' 010330'         MBUFF1: .WORD   BUFF1           ;THIS LOC USED TO GET THE PA OF  BUFFER1
000326' 000000          GBUFF1: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000330' 000000          EBUFF1: .WORD   0               ;CONTAINS EA BITS SHIFTED
000332' 012334'         MBUFF2: .WORD   BUFF2           ;THIS LOC USED TO GET PA ADRS OF BUFFER2
000334' 000000          GBUFF2: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000336' 000000          EBUFF2: .WORD   0               ;CONTAINS EA BITS
000340' 013336'         MBUFF3: .WORD   BUFF3           ;THIS LOC USED TO GET PA OF BUFFER3
000342' 000000          GBUFF3: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000344' 000000          EBUFF3: .WORD   0               ;CONTAINS EA BITS
000346' 005240'         MJOB0:  .WORD   JOB0            ;THIS LOC USDED TO GET PA OF JOB0
000350' 000000          GJOB0:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000352' 000000          EJOB0:  .WORD   0               ;CONTAINS EA BITS
000354' 005334'         MJOB1:  .WORD   JOB1            ;THIS LOC USDED TO GET PA OF JOB1
000356' 000000          GJOB1:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000360' 000000          EJOB1:  .WORD   0               ;CONTAINS EA BITS
000362' 005430'         MJOB2:  .WORD   JOB2            ;THIS LOC USDED TO GET PA OF JOB2
000364' 000000          GJOB2:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000366' 000000          EJOB2:  .WORD   0               ;CONTAINS EA BITS
000370' 005524'         MJOB3:  .WORD   JOB3            ;THIS LOC USDED TO GET PA OF JOB3
000372' 000000          GJOB3:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000374' 000000          EJOB3:  .WORD   0               ;CONTAINS EA BITS
000376' 005620'         MJOB4:  .WORD   JOB4            ;THIS LOC USDED TO GET PA OF JOB4
000400' 000000          GJOB4:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000402' 000000          EJOB4:  .WORD   0               ;CONTAINS EA BITS
000404' 005714'         MJOB5:  .WORD   JOB5            ;THIS LOC USDED TO GET PA OF JOB5
000406' 000000          GJOB5:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000410' 000000          EJOB5:  .WORD   0               ;CONTAINS EA BITS
000412' 006010'         MJOB6:  .WORD   JOB6            ;THIS LOC USDED TO GET PA OF JOB6
000414' 000000          GJOB6:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000416' 000000          EJOB6:  .WORD   0               ;CONTAINS EA BITS
000420' 006104'         MJOB7:  .WORD   JOB7            ;THIS LOC USDED TO GET PA OF JOB7
000422' 000000          GJOB7:  .WORD   0               ;CONTAINS PA (FROM MONITOR)
000424' 000000          EJOB7:  .WORD   0               ;CONTAINS EA BITS
000426' 005332'         MJOB0U: .WORD   JOB0U           ;THIS LOC USDED TO GET PA OF JOB0U
000430' 000000          GJOB0U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000432' 000000          EJOB0U: .WORD   0               ;CONTAINS EA BITS
000434' 005426'         MJOB1U: .WORD   JOB1U           ;THIS LOC USDED TO GET PA OF JOB1U
000436' 000000          GJOB1U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000440' 000000          EJOB1U: .WORD   0               ;CONTAINS EA BITS
000442' 005522'         MJOB2U: .WORD   JOB2U           ;THIS LOC USDED TO GET PA OF JOB2U
000444' 000000          GJOB2U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000446' 000000          EJOB2U: .WORD   0               ;CONTAINS EA BITS
000450' 005616'         MJOB3U: .WORD   JOB3U           ;THIS LOC USDED TO GET PA OF JOB3U
000452' 000000          GJOB3U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
```



```
000454' 000000          EJOB3U: .WORD   0               ;CONTAINS EA BITS
000456' 005712'         MJOB4U: .WORD   JOB4U           ;THIS LOC USDED TO GET PA OF JOB4U
000460' 000000          GJOB4U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000462' 000000          EJOB4U: .WORD   0               ;CONTAINS EA BITS
000464' 006006'         MJOB5U: .WORD   JOB5U           ;THIS LOC USDED TO GET PA OF JOB5U
000466' 000000          GJOB5U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000470' 000000          EJOB5U: .WORD   0               ;CONTAINS EA BITS
000472' 006102'         MJOB6U: .WORD   JOB6U           ;THIS LOC USDED TO GET PA OF JOB6U
000474' 000000          GJOB6U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000476' 000000          EJOB6U: .WORD   0               ;CONTAINS EA BITS
000500' 006176'         MJOB7U: .WORD   JOB7U           ;THIS LOC USDED TO GET PA OF JOB7U
000502' 000000          GJOB7U: .WORD   0               ;CONTAINS PA (FROM MONITOR)
000504' 000000          EJOB7U: .WORD   0               ;CONTAINS EA BITS


000506'                 START:
000506'                 RESTRT:
000506' 016767  177274  177514          MOV     ADDR,KMAD0      ;SET UP BASE ADDR.
                                        ;
                                        ;THIS SECTION OF CODE HANDLES INITIALIZING DPA-11 FUNCTIONS
                                        ;


000514' 016700  177510                  MOV     KMAD0,R0        ;GET KMC-11 ADDRESS.
000520' 012701  000232'                 MOV     #KMAD1,R1       ;GET ADDR. OF ADDR. LIST.

000524' 005200          64$:    INC     R0              ;UPDATE ADDR.
000526' 010021                  MOV     R0,(R1)+        ;WRITE ADDR.
000530' 020127  000250'         CMP     R1,#KMAD7+2     ;DONE ALL ADDRESSES?
000534' 001373                  BNE     64$             ;NO - DO NEXT ADDR.


                                ;IF DEDICATED MICRO-CODE WAS SELECTED
                                ;THE ENCODE SR2 VALUE BECOMES THE CLOCK PRESET VALUES
                                ;IF NOT USE THE DEFAULT RATE <ABOUT 10KHZ>

000536' 016700  177256          MOV     SR2,R0          ;GET CURRENT CODE
000542' 042700  177770          BIC     #177770,R0      ;MASK OFF OTHER BITS
000546' 006300                  ASL     R0              ;CONVERT INTO WORD OFFSET
000550' 016067  000560' 177450  MOV     10$(R0),SR2VAL  ;INDEX INTO TABLE OF PRESET VALUES
000556' 000410                  BR      1$              ;BR OVER TABLE OF VALUES
000560' 177763          10$:    .WORD   -13.            ;150K RATE
000562' 177761                  .WORD   -15.            ;134K
000564' 177757                  .WORD   -17.            ;118K
000566' 177754                  .WORD   -20.            ;100K
000570' 177747                  .WORD   -25.            ;80K
000572' 177737                  .WORD   -33.            ;60.6K
000574' 177716                  .WORD   -50.            ;40K
000576' 177635                  .WORD   -99.            ;20K
                                ;CHECK TO BE SURE SR1 HAS BEEN INITIALIZED BY THE OPERATOR
000600' 005767  177212  1$:     TST     SR1             ;HAS SR1 BEEN MODIFIED ?
000604' 001010                  BNE     TST1            ;YES-BRANCH TO START OF TESTS
000606' 012767  000007  177272  MOV     #7,ERRTYP       ;LOAD "SELECTION" ERROR CODE
000614' 104403  000000' 006200' MSGNS,BEGIN,WARNMSG     ;ASCII MESSAGE CALL WITH COMMON HEADER
000622' 104410  000000'         ENDS,BEGIN
                                ;
                                ;ABORT THIS MODULE SINCE WE HAVE NOT
                                ;BEEN INFORMED WHICH DEVICES ARE ON
                                ;THE I/O BUS VIA SR1
```

```
                                ;DETERMINE THE TYPE OF MICROCODE TO LOAD AND THEN LOAD AND VERIFY THE LOAD

000626' 032767  000100  177162 TST1:   BIT     #BIT6,SR1       ;TEST MICRO-CODE TYPE INDICATOR
000634' 001010                         BNE     1$              ;BR IF DEDICATED USER
000636' 012700  015346'                MOV     #MMAST,R0       ;GET ADDRESS OF MULTI-USER
000642' 010067  177434                 MOV     R0,UCODES       ;SAVE STARTING ADDRESS OF U CODE
000646' 012767  000115  177424         MOV     #'M,UCODE       ;LOAD TYPE OF UCODE
000654' 000407                         BR      2$              ;BR OVER SETUP
000656' 012700  021352'        1$:     MOV     #UMAST,R0       ;GET ADDRESS OF DEDICATED USER
000662' 010067  177414                 MOV     R0,UCODES       ;SAVE STATING ADDRESS
000666' 012767  000104  177404         MOV     #'D,UCODE       ;LOAD TYPE OF UCODE
000674' 005077  177330         2S:     CLR     @KMAD0          ;CLEAR CSR
000700' 005077  177334                 CLR     @KMAD4          ;CLEAR SCRAM ADDRESS
000704' 052777  002000  177316 3S:     BIS     #2000,@KMAD0    ;SELECT CRAM
000712' 012077  177326                 MOV     (R0)+,@KMAD6    ;WRITE DATA
000716' 052777  020000  177304         BIS     #20000,@KMAD0   ;SET CRAM WRITE
000724' 005077  177300                 CLR     @KMAD0          ;DISABLE CRAM
000730' 005277  177304                 INC     @KMAD4          ;UPDATE CRAM ADDRESS
000734' 021027  177777                 CMP     (R0),#-1        ;ALL DONE
000740' 001361                         BNE     3S              ;NO-LOOP
000742' 005077  177272                 CLR     @KMAD4          ;CLEAR CRAM ADDRESS
000746' 016700  177330                 MOV     UCODES,R0       ;GET MICRO-CODE ADDRESS
000752' 052777  002000  177250 4S:     BIS     #2000,@KMAD0    ;SELECT CRAM
000760' 022077  177260                 CMP     (R0)+,@KMAD6    ;DATA-OK?
000764' 001010                         BNE     5S              ;NO-REPORT ERROR
000766' 021027  177777                 CMP     (R0),#-1        ;ALL DONE
000772' 001431                         BEQ     TST2            ;YES EXIT
000774' 005077  177230                 CLR     @KMAD0          ;NO DESELECT CRAM
001000' 005277  177234                 INC     @KMAD4          ;UPDATE CRAM ADDRESS
001004' 000762                         BR      4S              ;BR AND CHECK NEXT WORD
                               ;SETUP CODE FOR ERROR REPORTING
001006' 016767  177232  177066 5S:     MOV     KMAD6,ACSR      ;STORE ADDRESS OF KMC-11
001014' 017767  177220  177062         MOV     @KMAD4,AMASADR  ;ADDRESS LOCATION OF FAILING MICRO-CODE
001022' 017767  177216  177060         MOV     @KMAD6,AMAS     ;ACTUAL MICRO-CODE LOADED INTO THE KMC
001030' 014067  177052                 MOV     -(R0),ASB       ;EXPECTED MICRO-CODE
001034' 010067  177042                 MOV     R0,SMADR        ;ADDRESS OF EXPECTED MICRO-CODE
001040' 012767  000052  177040         MOV     #52,ERRTYP      ;LOAD "MICRO-CODE" NOT LOADED ERROR CODE
                               ;*****************************************************************
001046' 104404  000000'                DATER$,BEGIN            ;DATA ERROR!!!
                               ;*****************************************************************
                               ;*
                               ;*THIS IS A FATAL KMC ERROR.
                               ;*THIS ERROR WAS DETECTED BY COMPARING THE KNOWN
                               ;*MICRO-CODE AGAINST THAT WHICH WAS LOADED
                               ;*"SMADR" CONTAINS THE ADDRESS OF WHAT THE MICRO-CODE SHOULD HAVE BEEN
                               ;*"WAS ADR" CONTAINS THE ADDRESS OF THE MICRO-CODE THAT WAS IN ERROR
                               ;*"AMAS" CONTAINS THE MICRO-CODE IN ERROR
                               ;*"ASB" CONTAINS WHAT THE MICRO-CODE SHOULD HAVE BEEN.
                               ;*DUE TO THE NATURE OF THIS ERROR, YOU CANNOT RUN "DXLPH"
                               ;*IT IS SUGGESTED THAT YOU RUN "MD-11-DRLPA".
001052' 000167  177430                 JMP     START           ;TRY STARTING AGAIN
```



```
                               ;TEST #2
                               ;IN THIS TEST WE WILL TRY TO START
                               ;THE USER MICRO-CODE PREVIOUSLY LOAD BY "TST1"
                               ;
001056'                 TST2:
001056' 052777  040000  177144         BIS     #BIT14,@KMAD0   ;SET INIT.
001064' 012701  000040                 MOV     #40,R1          ;LOAD DELAY COUNTER
001070'                 1S:
001070' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001074' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001100' 005301                         DEC     R1              ;DELAY
001102' 001372                         BNE     1S              ;  LOOP
001104' 012777  104000  177116         MOV     #BIT15!BIT11,@KMAD0     ;BIT 15 OF THE KMC CSR SHOULD BE
                                                                       ;SET AT THIS TIME.
001112' 012701  000500                 MOV     #320.,R1        ;LOAD DELAY COUNTER
001116' 032777  000200  177104 2S:     BIT     #BIT7,@LPCI     ;BIT 7 OF CONTROL IN REG. (READY IN)
001124' 001020                         BNE     3S              ;SHOULD BE SET AT INIT.
001126' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001132' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001136' 005301                         DEC     R1              ;DELAY
001140' 001366                         BNE     2S              ;  LOOP
001142' 012767  000034  176736         MOV     #34,ERRTYP      ;LOAD "DEVICE WILL NOT INIT" ERROR CODE
001150' 004767  005220                 JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
                               ;*****************************************************************
001154' 104405  000000' 000000         HRDER$,BEGIN,NULL       ;;LPA-11 ERROR - READY IN NOT SET AT INIT.
                               ;*****************************************************************
001162' 000167  177320                 JMP     START           ;TRY STARTING AGAIN

001166' 032777  000200  177040 3S:     BIT     #BIT7,@LPCI     ;CHECK READY OUT, IT SHOULD
001174' 001412                         BEQ     TST3            ;BE CLEAR AT INIT.
001176' 012767  000034  176702         MOV     #34,ERRTYP      ;LOAD "DEVICE WILL NOT INIT" ERROR CODE
001204' 004767  005164                 JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
                               ;*****************************************************************
001210' 104405  000000' 000000         HRDER$,BEGIN,NULL       ;;LPA-11 ERROR - READY OUT SET AT INIT.
                               ;*****************************************************************
00121A' 000167  177264                 JMP     START           ;TRY STARTING AGAIN
```

```
                                        ;TEST #3
                                        ;INIT TEST IN THIS TEST WE MAKE SURE THE KMC MICRO-
                                        ;CODE CAN FIND ALL ADDRESSES OF DEVICES TO BE TESTED.
                                ;NOW MAKE SURE A CLOCK IS SELECTED
001222' 012701  006472'         TST3:   MOV     #DEVLST+2,R1    ;PRIME POINTER TO THE "INIT" COMMAND
001226' 032767  000002  176562          BIT     #BIT1,SR1       ;KW11K CLOCK??
001234' 001403                          BEQ     1$              ;NO
001236' 016721  177012                  MOV     KW11K,(R1)+     ;YES FIX ADDR.
001242' 000432                          BR      TST3A
001244' 032767  002000  176544  1$:     BIT     #BIT10,SR1      ;AR11 CLOCK??
001252' 001405                          BEQ     2$              ;NO
001254' 016711  177014                  MOV     AR11K,(R1)      ;YES FIX AR11 CLOCK ADDR.
001260' 062721  000004                  ADD     #4,(R1)+
001264' 000421                          BR      TST3A
001266' 032767  020000  176522  2$:     BIT     #BIT13,SR1      ;LPS CLOCK?
001274' 001405                          BEQ     3$              ;BR IF NO CLOCK WAS SELECTED
001276' 016711  176774                  MOV     LPS11,(R1)      ;YES FIX LPS11 CLOCK ADDR.
001302' 062721  000004                  ADD     #4,(R1)+
001306' 000410                          BR      TST3A
001310' 012767  000007  176570  3$:     MOV     #7,ERRTYP       ;LOAD "SELECTION" ERROR CODE
                                        ;*****************************************************************
001316' 104405  000000' 000000          HRDERR$,BEGIN,NULL      ;;NO CLOCK SELECTED FOR TEST...
                                        ;*****************************************************************
                                ;CAN NOT DO ANY MORE TESTS WITHOUT CLOCK PRESENT! YOU MUST SELECT A CLOCK.
001324' 104410  000000'                 ENDS,BEGIN              ;
                                ;NOW DETERMINE IF ANOTHER DEVICE WAS SELECTED
001330' 032767  002000  176460  TST3A:  BIT     #BIT10,SR1      ;TEST IF AR11 SELECTED
001336' 001014                          BNE     TST3B           ;BR IF AR11
001340' 032767  155675  176450          BIT     #155675,SR1     ;TEST IF OTHER THAN A CLOCK WAS SELECTED
001346' 001010                          BNE     TST3B           ;BR IF OTHER THAN CLOCK ALSO SELECTED
001350' 012767  000007  176530          MOV     #7,ERRTYP       ;LOAD "SELECTION" ERROR CODE
                                        ;*****************************************************************
001356' 104405  000000' 000000          HRDERR$,BEGIN,NULL      ;;ONLY A CLOCK WAS SELECTED...
                                        ;*****************************************************************
001364' 104410  000000'                 ENDS,BEGIN              ;
                                ;NOW DECODE "SR1" FOR SYSTEM CONFIG.
                                ; AND BUILD INITILIZE COMMAND IN RDA FORMAT
001370' 012721  000001          TST3B:  MOV     #1,(R1)+        ;NO "CLOCK B" WILL BE USED
001374' 032767  000001  176414          BIT     #BIT0,SR1       ;AD11K FOR TEST??
001402' 001403                          BEQ     21$
001404' 016721  176642                  MOV     AD11K,(R1)+     ;YES FIX AD11K ADDR.
001410' 000420                          BR      30$
001412' 032767  002000  176376  21$:    BIT     #BIT10,SR1      ;AR11 A/D?
001420' 001403                          BEQ     22$
001422' 016721  176646                  MOV     AR11K,(R1)+
001426' 000411                          BR      30$
001430' 032767  010000  176360  22$:    BIT     #BIT12,SR1      ;LPS A/D?
001436' 001403                          BEQ     23$
001440' 016721  176632                  MOV     LPS11,(R1)+
001444' 000402                          BR      30$
001446' 012721  000001          23$:    MOV     #1,(R1)+        ;INDICATE NO "FIRST A TO D"
001452' 032767  000020  176336  30$:    BIT     #BIT4,SR1       ;2ND AD11K?
001460' 001403                          BEQ     31$
001462' 016721  176574                  MOV     AD11K2,(R1)+
001466' 000402                          BR      32$
001470' 012721  000001          31$:    MOV     #1,(R1)+        ;INDICATE NO "SECOND A TO D"
```



```
001474' 032767  000010  176314  32$:    BIT     #BIT3,SR1       ;AA11K?
001502' 001403                          BEQ     33$
001504' 016721  176550                  MOV     AA11K,(R1)+
001510' 000424                          BR      40$
001512' 032767  002000  176276  33$:    BIT     #BIT10,SR1      ;AR11?
001520' 001405                          BEQ     34$
001522' 016711  176546                  MOV     AR11K,(R1)
001526' 062721  000016                  ADD     #16,(R1)+
001532' 000413                          BR      40$
001534' 032767  040000  176254  34$:    BIT     #BIT14,SR1      ;LPS D/A?
001542' 001405                          BEQ     35$
001544' 016711  176526                  MOV     LPS11,(R1)
001550' 062721  000016                  ADD     #16,(R1)+
001554' 000402                          BR      40$
001556' 012721  000001          35$:    MOV     #1,(R1)+        ;INDICATE NO "D TO A"
001562' 032767  000004  176226  40$:    BIT     #BIT2,SR1       ;DR11K?
001570' 001403                          BEQ     41$
001572' 016721  176460                  MOV     DR11K1,(R1)+
001576' 000413                          BR      50$
001600' 032767  100000  176210  41$:    BIT     #BIT15,SR1      ;LPS I/O?
001606' 001405                          BEQ     42$
001610' 016711  176462                  MOV     LPS11,(R1)
001614' 062721  000010                  ADD     #10,(R1)+       ;FIX LPS11 I/O ADDR.
001620' 000402                          BR      50$
001622' 012721  000001          42$:    MOV     #1,(R1)+        ;INDICATE NO "1ST DIGITAL I/O"
001626' 032767  000040  176162  50$:    BIT     #BIT5,SR1       ;DR11K #2 ?
001634' 001403                          BEQ     51$
001636' 016721  176422                  MOV     DR11K2,(R1)+
001642' 000402                          BR      52$
001644' 012721  000001          51$:    MOV     #1,(R1)+        ;INDICATE NO "2ND DIGITAL I/O"
001650' 032767  000200  176140  52$:    BIT     #BIT7,SR1       ;DR11K #3 ?
001656' 001403                          BEQ     53$
001660' 016721  176402                  MOV     DR11K3,(R1)+
001664' 000402                          BR      54$
001666' 012721  000001          53$:    MOV     #1,(R1)+        ;INDICATE NO "3RD DIGITAL I/O"
001672' 032767  000400  176116  54$:    BIT     #BIT8,SR1       ;DR11K #4 ?
001700' 001403                          BEQ     55$
001702' 016721  176362                  MOV     DR11K4,(R1)+
001706' 000402                          BR      56$
001710' 012721  000001          55$:    MOV     #1,(R1)+        ;INDICATE NO "4TH DIGITAL I/O"
001714' 032767  001000  176074  56$:    BIT     #BIT9,SR1       ;DR11K #5 ?
001722' 001403                          BEQ     57$
001724' 016721  176342                  MOV     DR11K5,(R1)+
001730' 000402                          BR      60$
001732' 012711  000001          57$:    MOV     #1,(R1)         ;INDICATE NO "5TH DIGITAL I/O"
001736' 116767  176306  004525  60$:    MOVB    VERSN,DEVLST+1  ;SAVE THE "VERSION" OF MICRO-CODE USED IN THIS MODULE
001744' 105067  004520                  CLRB    DEVLST          ;ENSURE "INIT" CODE
```

```
                                ;NOW START THE "INITILIZE COMMAND" TO THE LPA-11
                                ;
001750' 012701  006470'         TST3C:  MOV     #DEVLST,R1      ;SET TO GET PA
001754' 010167  176330                  MOV     R1,GETADR       ;MOV ADRS SO WE CAN GET THE PA
001760' 104415  000000' 000310'         GETPAS,BEGIN, GETADR            ;GET PHYSICAL ADDRESS FROM 16-BIT GETADR
001766' 016701  176320                  MOV     GETPAH,R1       ;RETURN PHYSICAL ADRS.
001772' 010177  176242                  MOV     R1,@LPADL       ;SET INIT POINTER
001776' 006267  176312                  ASR     GETEAB          ;SHIFT EA BITS INTO BITS 2 AND 3
002002' 006267  176306                  ASR     GETEAB
002006' 142777  000014  176214          BICB    #14,@LPC1       ;REMOVE OLD "EA" BITS
002014' 156777  176274  176206          BISB    GETEAB,@LPC1    ;SET EA BITS
002022' 052777  000001  176200          BIS     #BIT0,@LPC1     ;SET GO!
002030' 005000                          CLR     R0              ;CLEAR "DELAY VALUE"
                                ;
002032'                         1$:
002032' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002036' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002042' 132777  000200  176160          BITB    #BIT7,@LPC1     ;DONE?
002050' 001020                          BNE     2$              ;YES1
002052' 105200                          INCB    R0              ;NO-WAIT
002054' 001366                          BNE     1$              ;BUT NOT TOO LONG...
002056' 004767  004312                  JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
002062' 122767  000326  176013          CMPB    #326,ACSR+1     ;TEST IF DEVICE NOT FOUND ERROR CODE
002070' 001425                          BEQ     3$              ;BR IF THAT ERROR CODE
002072' 012767  000034  176006          MOV     #34,ERRTYP      ;LOAD "DEVICE WILL NOT INITILIZE" ERROR CODE
                                        ;*****************************************************************
002100' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;;LPA FAILED TO FINISH INIT.
                                        ;*****************************************************************
002106' 000167  176374                  JMP     START           ;TRY STARTING AGAIN
                                ;BR HERE UPON "LPA ERROR FLAG" BEING SET
002112' 105777  176116          2$:     TSTB    @LPC0           ;DID CONTROL OUT SET WITH ERROR??
002116' 100022                          BPL     TST4            ;BR IF NO ERROR
002120' 004767  004250                  JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
002124' 012767  000044  175754          MOV     #44,ERRTYP      ;LOAD "FLAG SHOULD NOT BE SET" ERROR CODE
                                        ;*****************************************************************
002132' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;;LPA ERROR CODE - CHECK RETURNED ERROR CODE
                                        ;*****************************************************************
                                        ; AGAINST DOCUMENTATION.
002140' 000167  176342                  JMP     START           ;TRY STARTING AGAIN
                                ;BR HERE UPON "DEVICE NOT FOUND ON LPA I/O BUS" LPA ERROR CODE
002144' 012767  000006  175734  3$:     MOV     #6,ERRTYP       ;LOAD "DEVICE NON-EXISTENT" ERROR CODE
                                        ;*****************************************************************
002152' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;;DEVICE DOES NOT EXIST? CHECK VALUE OF SR1
                                        ;*****************************************************************
002160' 000167  176322                  JMP     START           ;TRY STARTING AGAIN
```



```
                                ;TEST 4, FIRST GET READY BY FIRST GETTING THE PHYSICAL ADDRESS
                                ;  OF ALL THE JOBS AND BUFFERS
                                ;
002164' 012767  007326' 176116  TST4:   MOV     #BUFFER,GETADR          ;SET UP TO GET ALL PA'S
002172' 104415  000000' 000310'         GETPAS,BEGIN, GETADR            ;GET PHYSICAL ADDRESS FROM 16-BIT GETADR
002200' 104415  000000' 000316'         GETPAS,BEGIN, MBUFF0            ;GET PHYSICAL ADDRESS FROM 16-BIT MBUFF0
002206' 104415  000000' 000324'         GETPAS,BEGIN, MBUFF1            ;GET PHYSICAL ADDRESS FROM 16-BIT MBUFF1
002214' 104415  000000' 000332'         GETPAS,BEGIN, MBUFF2            ;GET PHYSICAL ADDRESS FROM 16-BIT MBUFF2
002222' 104415  000000' 000340'         GETPAS,BEGIN, MBUFF3            ;GET PHYSICAL ADDRESS FROM 16-BIT MBUFF3
002230' 104415  000000' 000346'         GETPAS,BEGIN, MJOB0             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB0
002236' 104415  000000' 000354'         GETPAS,BEGIN, MJOB1             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB1
002244' 104415  000000' 000362'         GETPAS,BEGIN, MJOB2             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB2
002252' 104415  000000' 000370'         GETPAS,BEGIN, MJOB3             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB3
002260' 104415  000000' 000376'         GETPAS,BEGIN, MJOB4             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB4
002266' 104415  000000' 000404'         GETPAS,BEGIN, MJOB5             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB5
002274' 104415  000000' 000412'         GETPAS,BEGIN, MJOB6             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB6
002302' 104415  000000' 000420'         GETPAS,BEGIN, MJOB7             ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB7
002310' 104415  000000' 000426'         GETPAS,BEGIN, MJOB0U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB0U
002316' 104415  000000' 000434'         GETPAS,BEGIN, MJOB1U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB1U
002324' 104415  000000' 000442'         GETPAS,BEGIN, MJOB2U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB2U
002332' 104415  000000' 000450'         GETPAS,BEGIN, MJOB3U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB3U
002340' 104415  000000' 000456'         GETPAS,BEGIN, MJOB4U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB4U
002346' 104415  000000' 000464'         GETPAS,BEGIN, MJOB5U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB5U
002354' 104415  000000' 000472'         GETPAS,BEGIN, MJOB6U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB6U
002362' 104415  000000' 000500'         GETPAS,BEGIN, MJOB7U            ;GET PHYSICAL ADDRESS FROM 16-BIT MJOB7U
002370' 012700  002422'                 MOV     #TABLE0,R0      ;LOAD POINTER TO TABLE
002374' 005710                  1$:     TST     (R0)            ;TEST IF AT END OF TABLE
002376' 001426                          BEQ     TST4A           ;BR IF DONE ALL ENTRIES
002400' 011002                          MOV     (R0),R2         ;GET AN ADDRESS VALUE
002402' 011201                          MOV     (R2),R1         ;GET THE VALUE
002404' 006201                          ASR     R1              ;MOVE THE DATA
002406' 006201                          ASR     R1              ;MOVE THE DATA
002410' 006201                          ASR     R1              ;MOVE THE DATA
002412' 006201                          ASR     R1              ;MOVE THE DATA
002414' 010112                          MOV     R1,(R2)         ;RELOAD THE NEW VALUE
002416' 005720                          TST     (R0)+           ;BUMP THE POINTER
002420' 000765                          BR      1$              ;BR AND DO ANOTHER TABLE ENTRY
                                ;
002422' 000322'                 TABLE0: EBUFF0                  ;EA BITS FOR EACH BUFFER ADDRESS
002424' 000330'                         EBUFF1
002426' 000336'                         EBUFF2
002430' 000344'                         EBUFF3
002432' 000432'                         EJOB0U                  ;EA BITS FOR EACH JOB "USW"
002434' 000440'                         EJOB1U
002436' 000446'                         EJOB2U
002440' 000454'                         EJOB3U
002442' 000462'                         EJOB4U
002444' 000470'                         EJOB5U
002446' 000476'                         EJOB6U
002450' 000504'                         EJOB7U
002452' 000000                          0
```

```
                                        ;
                                        ;HERE WE'LL EXERCISE ALL DEVICES THAT
                                        ;THE USER INDICATES HE HAS ON THE SYSTEM
                                        ;IF MULTI-USER MICRO CODE IS LOADED,
                                        ;IF DEDICATED USER MICRO CODE IS LOADED,
                                        ;WE'LL ONLY EXERCISE THE A/DS.

002454' 012767  177600  003710  TST4A:  MOV     #177600,K%TPRE  ;SET AN INITIAL CLOCK RATE PRESET.
002462' 012700  005240'         1$:     MOV     #JOB0,R0        ;PRIME THE POINTER TO BUFFER
002466' 005020                  1$:     CLR     (R0)+           ;CLEAR JOB ENTRY
002470' 020027  006200'                 CMP     R0,#WARNMSG     ;TEST IF DONE CLEARING
002474' 001374                          BNE     1$              ;BR IF NOT
002476' 005067  175604                  CLR     $GDDAT          ;CLEAR "# OF JOBS STARTED" COUNTER

                                ;DETERMINE THE TYPE OF "JOBS" AND LOAD THE "DATA TRANSFER START RDA FORMAT"

                                ;DETERMINE IF JOB 0 (FIRST A TO D) IS TO BE RUN

002502' 032767  012001  175306  DJOB0:  BIT     #BIT0!BIT10!BIT12,SR1 ;ANY 1ST A/D'S?
002510' 001510                          BEQ     DJOB1           ;NO-EXIT 1ST A/D'S.
002512' 012700  005240'                 MOV     #JOB0,R0        ;UPDATE JOB POINTER
002516' 012710  000012                  MOV     #12,(R0)        ;MODE WORD=START A/D.
002522' 052710  000400                  BIS     #BIT8,(R0)      ;SINGLE CHAN.
002526' 012760  000512  000002          MOV     #512,2(R0)      ;TAKE 512 SAMPLES
002534' 005067  002572                  CLR     JOB0U           ;CLEAR USW
002540' 016760  175664  000004          MOV     GJOB0U,4(R0)    ;LOAD PA OF USW
002546' 112760  000202  000007          MOVB    #BIT7!BIT1,7(0) ;BUFFER OVERRUN NOT FATAL, THREE BUFFERS.
002554' 016760  175540  000010          MOV     GBUFF0,10(R0)   ;SET BUFFER ADDR.
002562' 016760  175532  000014          MOV     GBUFF0,14(R0)
002570' 016760  175524  000020          MOV     GBUFF0,20(R0)
002576' 012760  000010  000054          MOV     #10,54(R0)      ;SET DELAY BEFORE START COUNT.
002604' 012760  000400  000056          MOV     #400,56(R0)     ;START CH0, INC=1
002612' 012760  000016  000060          MOV     #16,60(R0)      ;SAMPLE 14 CHANNELS
002620' 012760  000200  000062          MOV     #200,62(R0)     ;SET DWELL VALUE

002626' 116760  175600  000006          MOVB    EJOB0U,6(R0)    ;NOW WE MUST SET ALL "EA" BITS
002634' 016760  175462  000012          MOV     EBUFF0,12(R0)
002642' 016760  175454  000016          MOV     EBUFF0,16(R0)
002650' 016760  175446  000022          MOV     EBUFF0,22(R0)
002656' 005267  175424                  INC     $GDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER

                                ;DETERMINE IF OPERATOR SELECTED "DEDICATED" FIRMWARE

002662' 126727  175412  000104          CMPB    UCODE,#'D       ;DEDICATED MICRO CODE SELECTED ?
002670' 001020                          BNE     DJOB1           ;BR IF NOT (MUST BE MULTI USER)

002672' 042710  000010                  BIC     #BIT3,(R0)      ;CLEAR MULTIUSER BIT.
002676' 032767  000020  175112          BIT     #BIT4,SR1       ;SECOND A/D?
002704' 001402                          BEQ     2$              ;NO CONTINUE.
002706' 012710  004442                  MOV     #4442,(R0)      ;YES=MAKE OPERATE IN DUAL A/D MODE.
002712' 012760  001520  000002  2$:     MOV     #1520,2(R0)     ;LARGER BUFFER.
002720' 016767  175302  003444          MOV     SR2VAL,K%TPRE   ;SET A HIGHER CLOCK RATE.
                                                                ;WE WILL TAKE SAMPLES AT (SR2) RATE
002726' 000167  001416                  JMP     ELOOP           ;BYPASS OTHER JOB SETUP
```



```
                                ;DETERMINE IF JOB 1 (SECOND A/D <MULTI USER MICRO-CODE SELECTED>) IS TO BE RUN

002732' 032767  000020  175056  DJOB1:  BIT     #BIT4,SR1       ;SECOND A/D?
002740' 001502                          BEQ     DJOB2           ;BR IF NO SECOND A/D
002742' 012700  005334'                 MOV     #JOB1,R0        ;UPDATE JOB STORAGE ARE POINTER.
002746' 012710  000012                  MOV     #12,(R0)        ;MODE WORD=START A/D.
002752' 052710  000400                  BIS     #BIT8,(R0)      ;SINGLE CHANNEL
002756' 012760  000512  000062          MOV     #512,2(R0)      ;TAKE 512 SAMPLES

002764' 005067  002436                  CLR     JOB1U           ;CLEAR USW
002770' 016760  175442  000004          MOV     GJOB1U,4(R0)    ;LOAD THE USW PA
002776' 112760  000202  000007          MOVB    #BIT7!BIT1,7(0) ;BUFFER OVERRUN NOT FATAL, ONE BUFFER.
003004' 016760  175316  000010          MOV     GBUFF1,10(R0)   ;SET BUFFER ADDR.
003012' 016760  175310  000014          MOV     GBUFF1,14(R0)
003020' 016760  175302  000020          MOV     GBUFF1,20(R0)
003026' 012760  000010  000054          MOV     #10,54(R0)      ;SET DELAY BEFORE START.
003034' 012760  000500  000056          MOV     #500,56(R0)     ;START CHANNEL = 64, INC = 1
003042' 012760  000016  000060          MOV     #16,60(R0)      ;SAMPLE 14 CHANNELS.
003050' 012760  000200  000062          MOV     #200,62(R0)     ;LOAD THE DWELL VALUE
003056' 005267  175224                  INC     $GDDAT          ;UPDATE # OF JOBS STARTED COUNTER
                                ;NOW LOAD THE "EA" BITS FOR THE BUFFER ADDRESSES
003062' 116760  175352  000006          MOVB    EJOB1U,6(R0)
003070' 016760  175234  000012          MOV     EBUFF1,12(R0)
003076' 016760  175226  000016          MOV     EBUFF1,16(R0)
003104' 016760  175220  000022          MOV     EBUFF1,22(R0)

                                ;DETERMINE IF OPERATOR SELECTED "DEDICATED" FIRMWARE

003112' 126727  175162  000104          CMPB    UCODE,#'D       ;DEDICATED MICRO CODE SELECTED ?
003120' 001012                          BNE     DJOB2           ;BR IF NOT (MUST BE MULTI USER)

003122' 042710  000010                  BIC     #BIT3,(R0)      ;CLEAR MULTI USER BIT
003126' 012760  001520  000002          MOV     #1520,2(R0)     ; ENABLE A LARGER BUFFER
003134' 016767  175066  003230          MOV     SR2VAL,K%TPRE   ;SET THE SELECTED CLOCK RATE
003142' 000167  001202                  JMP     ELOOP           ;BYPASS OTHER JOB SETUP
```

```
                                        ;DETERMINE IF JOB 2 (D TO A) IS TO BE RUN
                                        ;
003146' 032767  042010  174642  DJOB2:  BIT     #BIT3!BIT10!BIT14,SR1 ;D/A OUTPUT SELECTED?
003154' 001457                          BEQ     DJOB3           ;NO-NEXT SET-UP
003156' 012700  005430'                 MOV     #JOB2,R0        ;UPDATE JOB STORAGE ARE POINTER
003162' 012710  001212                  MOV     #BIT12!BIT3!BIT7!BIT9,(R0) ;START, D/A, SEQ. CHAN.
003166' 012760  000256  000002          MOV     #256.,2(R0)     ;SET XFERR COUNT
                                        ;
003174' 005067  002322                  CLR     JOB2U           ;CLEAR USW
003200' 016760  175240  000004          MOV     GJOB2U,4(R0)    ;LOAD THE USW PA INTO THE RDA
003206' 112760  000202  000007          MOVB    #BIT7!BIT1,7(0) ;3 BUFFERS.
003214' 016760  175122  000010          MOV     GBUFF3,10(R0)   ;BUFFER ADDR.
003222' 016760  175114  000014          MOV     GBUFF3,14(R0)
003230' 016760  175106  000020          MOV     GBUFF3,20(R0)
003236' 012760  000100  000054          MOV     #100.,54(R0)    ;DELAY BEFORE START COUNTER.
003244' 012760  000001  000060          MOV     #1,60(R0)
003252' 012760  000040  000062          MOV     #40,62(R0)      ;LOAD DWELL VALUE
003260' 005267  175072                  INC     SGDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
                                        ;NOW LOAD THE "EA" BITS FOR THE BUFFER ADDRESSES
003264' 116760  175156  000006          MOVB    EJOB2U,6(R0)    ;SET EA BITS
003272' 016760  175046  000012          MOV     EBUFF3,12(R0)
003300' 016760  175040  000016          MOV     EBUFF3,16(R0)
003306' 016760  175032  000022          MOV     EBUFF3,22(R0)
                                        ;DETERMINE IF JOB 3 (DR11K #1 OR LPSDR) IS TO BE RUN
003314'                                 DJOB3:
                                        ;
003314' 032767  100004  174474          BIT     #BIT12!BIT15,SR1 ;/THIS DR11K #1 OR LPS I/O SEL?
003322' 001462                          BEQ     64$
003324' 012700  005524'                 MOV     #JOB3,R0        ;/UPDAT JOB STORAGE AREA POINTER.
003330' 012710  000432                  MOV     #432,(R0)       ;/SET I/O START, MULTI-USER, DIGITAL INPUT, SINGLE CHAN.
003334' 012760  000256  000002          MOV     #256,2(R0)      ;/WORD COUNT
                                        ;
003342' 005067  002250                  CLR     JOB3U           ;CLEAR USW
003346' 016760  175100  000004          MOV     GJOB3U,4(R0)    ;LOAD PA OF USW INTO RDA
003354' 112760  000202  000007          MOVB    #BIT7!BIT1,7(R0)        ;3 BUFFERS.
003362' 016760  174746  000010          MOV     GBUFF2,10(R0)   ;/SET BUFFER ADDR.
003370' 016760  174740  000014          MOV     GBUFF2,14(R0)
003376' 016760  174732  000020          MOV     GBUFF2,20(R0)
003404' 005267  174676                  INC     SGDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
003410' 012760  000100  000054          MOV     #100,54(R0)     ;DELAY BEFORE START.
003416' 012760  000000  000056          MOV     #1-1,56(R0)     ;/SELECT CHAN #.
003424' 012760  000001  000060          MOV     #1,60(R0)       ;/SAMPLING ONLY ONE CHAN.
003432' 012760  000040  000062          MOV     #40,62(R0)      ;SET THE DWELL VALUE
                                        ;NOW LOAD THE "EA" BITS FOR THE ADDRESSES
003440' 116760  175010  000006          MOVB    EJOB3U,6(R0)
003446' 016760  174664  000012          MOV     EBUFF2,12(R0)
003454' 016760  174656  000016          MOV     EBUFF2,16(R0)
003462' 016760  174650  000022          MOV     EBUFF2,22(R0)
003470'                                 64$:
```



```
                                        ;DETERMINE IF JOB 4 (DR11K #2) IS TO BE RUN
003470'                                 DJOB4:
                                        ;
003470' 032767  000040  174320          BIT     #BIT5,SR1       ;/THIS DR11K #2 SELECTED?
003476' 001462                          BEQ     64$
003500' 012700  005620'                 MOV     #JOB4,R0        ;/UPDAT JOB STORAGE AREA POINTER.
003504' 012710  000432                  MOV     #432,(R0)       ;/SET I/O START, MULTI-USER, DIGITAL INPUT, SINGLE CHAN.
003510' 012760  000256  000002          MOV     #256,2(R0)      ;/WORD COUNT
                                        ;
003516' 005067  002170                  CLR     JOB4U           ;CLEAR USW
003522' 016760  174732  000004          MOV     GJOB4U,4(R0)    ;LOAD PA OF USW INTO RDA
003530' 112760  000202  000007          MOVB    #BIT7!BIT1,7(R0)        ;3 BUFFERS.
003536' 016760  174572  000010          MOV     GBUFF2,10(R0)   ;/SET BUFFER ADDR.
003544' 016760  174564  000014          MOV     GBUFF2,14(R0)
003552' 016760  174556  000020          MOV     GBUFF2,20(R0)
003560' 005267  174522                  INC     SGDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
003564' 012760  000100  000054          MOV     #100,54(R0)     ;DELAY BEFORE START.
003572' 012760  000001  000056          MOV     #2-1,56(R0)     ;/SELECT CHAN #.
003600' 012760  000001  000060          MOV     #1,60(R0)       ;/SAMPLING ONLY ONE CHAN.
003606' 012760  000040  000062          MOV     #40,62(R0)      ;SET THE DWELL VALUE
                                        ;NOW LOAD THE "EA" BITS FOR THE ADDRESSES
003614' 116760  174642  000006          MOVB    EJOB4U,6(R0)
003622' 016760  174510  000012          MOV     EBUFF2,12(R0)
003630' 016760  174502  000016          MOV     EBUFF2,16(R0)
003636' 016760  174474  000022          MOV     EBUFF2,22(R0)
003644'                                 64$:
                                        ;DETERMINE IF JOB 5 (DR11K #3) IS TO BE RUN
003644'                                 DJOB5:
                                        ;
003644' 032767  000200  174144          BIT     #BIT7,SR1       ;/THIS DR11K #3 SELECTED?
003652' 001462                          BEQ     64$
003654' 012700  005714'                 MOV     #JOB5,R0        ;/UPDAT JOB STORAGE AREA POINTER.
003660' 012710  000432                  MOV     #432,(R0)       ;/SET I/O START, MULTI-USER, DIGITAL INPUT, SINGLE CHAN.
003664' 012760  000256  000002          MOV     #256,2(R0)      ;/WORD COUNT
                                        ;
003672' 005067  002110                  CLR     JOB5U           ;CLEAR USW
003676' 016760  174564  000004          MOV     GJOB5U,4(R0)    ;LOAD PA OF USW INTO RDA
003704' 112760  000202  000007          MOVB    #BIT7!BIT1,7(R0)        ;3 BUFFERS.
003712' 016760  174416  000010          MOV     GBUFF2,10(R0)   ;/SET BUFFER ADDR.
003720' 016760  174410  000014          MOV     GBUFF2,14(R0)
003726' 016760  174402  000020          MOV     GBUFF2,20(R0)
003734' 005267  174346                  INC     SGDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
003740' 012760  000100  000054          MOV     #100,54(R0)     ;DELAY BEFORE START.
003746' 012760  000002  000056          MOV     #3-1,56(R0)     ;/SELECT CHAN #.
003754' 012760  000001  000060          MOV     #1,60(R0)       ;/SAMPLING ONLY ONE CHAN.
003762' 012760  000040  000062          MOV     #40,62(R0)      ;SET THE DWELL VALUE
                                        ;NOW LOAD THE "EA" BITS FOR THE ADDRESSES
003770' 116760  174474  000006          MOVB    EJOB5U,6(R0)
003776' 016760  174334  000012          MOV     EBUFF2,12(R0)
004004' 016760  174326  000016          MOV     EBUFF2,16(R0)
004012' 016760  174320  000022          MOV     EBUFF2,22(R0)
004020'                                 64$:
```

```
                                        ;DETERMINE IF JOB 6 (DR11K #4) IS TO BE RUN
004020'                                 DJOB6:
                                        
004020' 032767  000400  173770          BIT     #BIT8,SR1       ;/THIS DR11K #4 SELECTED?
004026' 001462                          BEQ     64$
004030' 012700  006010'                 MOV     #JOB6,R0        ;/UPDAT JOB STORAGE AREA POINTER.
004034' 012710  000432                  MOV     #432,(R0)       ;/SET I/O START, MULTI-USER, DIGITAL INPUT, SINGLE CHAN.
004040' 012760  000256  000002          MOV     #256,2(R0)      ;/WORD COUNT
                                        
004046' 005067  002030                  CLR     JOB6U           ;CLEAR USW
004052' 016760  174416  000004          MOV     GJOB6U,4(R0)    ;LOAD PA OF USW INTO RDA
004060' 112760  000202  000007          MOVB    #BIT7!BIT1,7(R0)        ;3 BUFFERS.
004066' 016760  174242  000010          MOV     GBUFF2,10(R0)   ;/SET BUFFER ADDR.
004074' 016760  174234  000014          MOV     GBUFF2,14(R0)
004102' 016760  174226  000020          MOV     GBUFF2,20(R0)
004110' 005267  174172                  INC     $GDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
004114' 012760  000100  000054          MOV     #100,54(R0)     ;DELAY BEFORE START.
004122' 012760  000003  000056          MOV     #4-1,56(R0)     ;/SELECT CHAN #.
004130' 012760  000001  000060          MOV     #1,60(R0)       ;/SAMPLING ONLY ONE CHAN.
004136' 012760  000040  000062          MOV     #40,62(R0)      ;SET THE DWELL VALUE
                                        ;NOW LOAD THE "EA" BITS FOR THE ADDRESSES
004144' 116760  174326  000006          MOVB    EJOB6U,6(R0)
004152' 016760  174160  000012          MOV     EBUFF2,12(R0)
004160' 016760  174152  000016          MOV     EBUFF2,16(R0)
004166' 016760  174144  000022          MOV     EBUFF2,22(R0)
004174'                                 64$:
                                        ;DETERMINE IF JOB 7 (DR11K #5) IS TO BE RUN
004174'                                 DJOB7:
                                        
004174' 032767  001000  173614          BIT     #BIT9,SR1       ;/THIS DR11K #5 SELECTED?
004202' 001462                          BEQ     64$
004204' 012700  006104'                 MOV     #JOB7,R0        ;/UPDAT JOB STORAGE AREA POINTER.
004210' 012710  000432                  MOV     #432,(R0)       ;/SET I/O START, MULTI-USER, DIGITAL INPUT, SINGLE CHAN.
004214' 012760  000256  000002          MOV     #256,2(R0)      ;/WORD COUNT
                                        
004222' 005067  001750                  CLR     JOB7U           ;CLEAR USW
004226' 016760  174250  000004          MOV     GJOB7U,4(R0)    ;LOAD PA OF USW INTO RDA
004234' 112760  000202  000007          MOVB    #BIT7!BIT1,7(R0)        ;3 BUFFERS.
004242' 016760  174066  000010          MOV     GBUFF2,10(R0)   ;/SET BUFFER ADDR.
004250' 016760  174060  000014          MOV     GBUFF2,14(R0)
004256' 016760  174052  000020          MOV     GBUFF2,20(R0)
004264' 005267  174016                  INC     $GDDAT          ;UPDATE "# OF JOBS STARTED" COUNTER
004270' 012760  000100  000054          MOV     #100,54(R0)     ;DELAY BEFORE START.
004276' 012760  000004  000056          MOV     #5-1,56(R0)     ;/SELECT CHAN #.
004304' 012760  000001  000060          MOV     #1,60(R0)       ;/SAMPLING ONLY ONE CHAN.
004312' 012760  000040  000062          MOV     #40,62(R0)      ;SET THE DWELL VALUE
                                        ;NOW LOAD THE "EA" BITS FOR THE ADDRESSES
004320' 116760  174160  000006          MOVB    EJOB7U,6(R0)
004326' 016760  174004  000012          MOV     EBUFF2,12(R0)
004334' 016760  173776  000016          MOV     EBUFF2,16(R0)
004342' 016760  173770  000022          MOV     EBUFF2,22(R0)
004350'                                 64$:
```



```
                                        ;NOW THAT THE "JOBS" HAVE BEEN BUILT,
                                        ;  BUILD THE "TO DO" QUEUE
                                        
                                                ;*      JOB0-JOB7 PRE SET UP
                                                ;*      $GDDAT=  NO. OF JOBS TO EXERCISE
                                                ;*
                                        
004350' 012700  007060'                 BLOOP:  MOV     #TODOQ,R0
004354' 005020                          1$:     CLR     (R0)+           ;CLEAR "TO DO" QUEUE AREA.
004356' 020027  007324'                         CMP     R0,#DONEC
004362' 001374                                  BNE     1$
004364' 012701  007060'                         MOV     #TODOQ,R1       ;GET QUES AREA
004370' 012702  005240'                         MOV     #JOB0,R2        ;PICK UP 1ST JOB ADDRESS
004374' 012703  005334'                         MOV     #JOB1,R3        ;CALCULATE ADDR DIF.
004400' 162703  005240'                         SUB     #JOB0,R3        ;  BETWEEN THE JOBS
004404' 016767  173676  002566                  MOV     $GDDAT,TODOC    ;REMEMBER HOW MANY JOBS ARE QUEUED
004412' 012700  000010                          MOV     #8.,R0          ;LOAD MAX JOB COUNT
004416' 005712                          2$:     TST     (R2)            ;ANYTHING IN THIS JOB??
004420' 001401                                  BEQ     3$
004422' 010221                                  MOV     R2,(R1)+        ;STORE JOB ADDRESS INTO "TO DO" QUEUE
004424' 060302                          3$:     ADD     R3,R2           ;POINT TO NEXT JOB
004426' 005300                                  DEC     R0              ;DONE ALL JOBS?
004430' 001372                                  BNE     2$              ;NO-LOOP.
                                        ;NOW LOAD THE CLOCK START RDA
004432' 012701  006366'                         MOV     #RWT,R1         ;SET UP TO GET PA
004436' 010167  173646                          MOV     R1,GETADR       ;MOV ADRS SO WE CAN GET THE PA
004442' 104415  000000' 000310'                 GETPAS,BEGIN, GETADR            ;GET PHYSICAL ADDRESS FROM 16-BIT GETADR
004450' 016701  173636                          MOV     GETPAR,R1       ;RETURN PHYSICAL ADRS.
004454' 010177  173560                          MOV     R1,@LPADL       ;START CLOCK
                                        
004460' 006267  173630                          ASR     GETEAB          ;MOVE EA BITS
004464' 006267  173624                          ASR     GETEAB          ;OVER TO BIT 2-3
004470' 042777  000014  173532                  BIC     #14,@LPCI       ;REMOVE OLD "EA" BITS
004476' 056777  173612  173524                  BIS     GETEAB,@LPCI    ;SET NEW "EA" BITS.
004504' 005067  173574                          CLR     $BDDAI          ;CLEAR "JOB STARTED" COUNTER
004510' 052777  000001  173512                  BIS     #BIT0,@LPCI     ;ENABLE THE LPA-11 TO START
                                        ;NOW PRIME THE LPA-11 INTERRUPT VECTORS
004516' 016700  173266                          MOV     VECTOR,R0
004522' 012720  004704'                         MOV     #OUPSRV,(R0)+
004526' 016720  173260                          MOV     BR1,(R0)+
004532' 012720  004562'                         MOV     #INPSRV,(R0)+
004536' 016720  173250                          MOV     BR1,(R0)+
004542' 052777  000100  173460                  BIS     #BIT6,@LPCI     ;ENABLE LPA-11 "INPUT" INTERRUPTS
004550' 052777  000100  173456                  BIS     #BIT6,@LPCO     ;ENABLE LPA-11 "OUTPUT" INTERRUPTS
004556' 104400  000000'                         EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                ;RETURN HERE UPON AN "INPUT" INTERRUPT FROM THE LPA-11
004562' 005767  002412          INPSRV: TST     TODOC           ;FINISHED ALL "TO DO" JOB REQUESTS
004566' 001445                          BEQ     IEX             ;BR IF DONE
                                        ;-----------------------------------------------------------------
004570' 000004  000000' 004576'         PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
004576' 012700  007060'         1$:     MOV     #TODOQ,R0       ;LOAD POINTER TO "TO DO" QUEUE
004602' 005720                  2$:     TST     (R0)+           ;FIND AN ENTRY
004604' 001776                          BEQ     2$
004606' 014067  173476                  MOV     -(R0),GETADR    ;SET TO GET PA
004612' 104415  000000' 000310'         GETPAS,BEGIN, GETADR            ;GET PHYSICAL ADDRESS FROM 16-BIT GETADR
004620' 016777  173466  173412          MOV     GETPAR,@LPADL   ;LOAD RDA OF THIS JOB INTO LPA11
004626' 006267  173462                  ASR     GETEAB
004632' 006267  173456                  ASR     GETEAB
004636' 042777  000014  173364          BIC     #14,@LPCI       ;REMOVE OLD "EA" BITS
004644' 056777  173444  173356          BIS     GETEAB,@LPCI    ;SET EXTION BITS
004652' 005010                          CLR     (R0)            ;CLEAR THIS JOB OUT OF THE QUEUE
004654' 005367  002320                  DEC     TODOC           ;REDUCE "TO DO" JOB COUNTER
004660' 001003                          BNE     3$              ;BR IF MORE REMAIN
004662' 042777  000100  173340          BIC     #BIT6,@LPCI     ;DON'T ALLOW ANOTHER INTERRUPT.
004670' 052777  000001  173332  3$:     BIS     #BIT0,@LPCI     ;SET GO!
004676' 104400  000000'                 EXITS,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
004702' 000002                  IEX:    RTI                     ;EXIT IF UNEXPECTED "INPUT" INTERRUPT

                                ;RETURN HERE UPON AN "OUTPUT" INTERRUPT FROM THE LPA-11
004704'                         OUPSRV:
                                        ;-----------------------------------------------------------------
004704' 000004  000000' 004712'         PIRQS,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                        ;-----------------------------------------------------------------
004712' 017700  173316          1$:     MOV     @LPCD,R0        ;NOW LETS GET USER NUMBER.
004716' 042700  177770                  BIC     #^C<7>,R0       ;RID OF REST OF JUNK.
004722' 006300                          ASL     R0              ;MAKE A WORD INDEX
004724' 017701  173310                  MOV     @LPADL,R1       ;READ THE "RDA" ADDRESS FROM THE LPA
004730' 105777  173302                  TSTB    @LPSD           ;SEE IF BIT15 SET SHOWING ERROR
004734' 100032                          BPL     NER             ;IF NO ERROR, BR AND PROCEED.
004736' 122777  000250  173272          CMPB    #250,@LPSD      ;JOB DONE <REQUEST TERMINATED> ERROR?
004744' 001014                          BNE     2$              ;NO-BR AND REPORT ERROR
004746' 005267  173332                  INC     $NODAT          ;UPDATE COUNTER OF "JOB DONE" ERRORS
004752' 026767  173330  173324          CMP     $GDDAT,$NODAT   ;COMPARE COUNT OF JOB DONE <REQUEST TERMINATED>
                                                                ; TO THE NUMBER OF JOBS STARTED
004760' 001122                          BNE     OUTCON          ;BR IF NOT EQUAL
004762' 004767  001406                  JSR     PC,RESET0       ;STOP THE LPA-11K
004766' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                ;MONITOR SHALL TEST END OF PASS
004772' 000167  173510                  JMP     START           ;RETURN TO TESTING
                                ;BR HERE UPON UNEXPECTED ERROR FLAG CONDITIONS
004776'                         2$:
004776' 004767  001372                  JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
005002' 012767  000000  173076          MOV     #0,ERRTYP       ;LOAD "NOT DEFINED" ERROR CODE
                                        ;*****************************************************************
005010' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;;LPA-11 MICRO CODE REPORTED AN ERROR
                                        ;*****************************************************************
005016' 000167  173464                  JMP     START           ;TRY STARTING AGAIN
```



```
                                ;COME HERE IF "OUT" INTERRUPT "ERROR" WAS NOT SET
005022' 001030                  NER:    BNE     3$              ;IF STATUS IS 1 OR 2 BR. IF =0 THEN
005024' 017702  173210                  MOV     @LPADL,R2       ;GET THE PA OF THE RDA FROM THE LPA
005030' 012703  000350'                 MOV     #GJOB0,R3       ;AND GO THRU THE TABLES TO FIND THE VA
005034' 020213                  1$:     CMP     R2,(R3)         ;DOES THE PA MATCH ?
005036' 001417                          BEQ     2$              ;YES GET THE VA FROM THE TABLE
005040' 062703  000006                  ADD     #6,R3           ;NO-UPDATE POINTER AND TRY AGAIN
005044' 022703  000436'                 CMP     #GJOB10,R3      ;TEST IF AT END OF LIST
005050' 001371                          BNE     1$              ;BR IF NOT
005052' 004767  001316                  JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
005056' 012767  000000  173022          MOV     #0,ERRTYP       ;LOAD "NOT DEFINED" ERROR CODE
                                        ;*****************************************************************
005064' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;LPA-11 RETURNED INCORRECT RDA ADDRESS
                                        ;*****************************************************************
005072' 000167  173410                  JMP     START           ;TRY STARTING AGAIN
005076' 016360  177776  007202' 2$:     MOV     -2(R3),TABLE(R0)        ;STORE THE VA OF THE RDA IN A TABLE
                                        ;   INDEXING OFF THE UID CODE RETURNED BY THE LPA MICRO-CODE

                                ;BR TO HERE UPON CODE 1 (BUFFER FULL/EMPTY) OR 2 (NONFATAL BUFFER OVERRUN)
                                ;FALL THROUGH IF CODE 0

005104' 016001  007202'         3$:     MOV     TABLE(R0),R1    ;REFETCH RDA VA ADDRS
005110' 016103  000004                  MOV     4(R1),R3        ;GET THE PA OF THE USW
005114' 012704  000430'                 MOV     #GJOB00,R4      ;SET TO FIND PA IN TABLE
005120' 020314                  4$:     CMP     R3,(R4)         ;DOES THE PA MATCH
005122' 001420                          BEQ     5$              ;YES-BR AND GET VA FROM TABLE
005124' 062704  000006                  ADD     #6,R4           ;NO-TRY NEXT ENTRY
005130' 022704  000510'                 CMP     #GJOB70+6,R4    ;TEST IF END OF BUFFER
005134' 001371                          BNE     4$              ;BR IF NOT
005136' 000240                          NOP
005140' 004767  001230                  JSR     PC,RESET0       ;GO AND READ LPA-11 REGISTERS
005144' 012767  000000  172734          MOV     #0,ERRTYP       ;LOAD "NOT DEFINED" ERROR CODE
                                        ;*****************************************************************
005152' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;LPA RETURNED INCURRECT USER NUMBER
                                        ;*****************************************************************
005160' 000167  173322                  JMP     START           ;TRY STARTING AGAIN

                                ;COME HERE UPON FINDING AN ADDRESS MATCH

005164' 016403  177776          5$:     MOV     -2(R4),R3       ;GET VA OF THE USW
005170' 116302  000001                  MOVB    1(R3),R2        ;GET THE LAST BUFFER USED
005174' 005202                          INC     R2              ;UPDATE FOR NEXT BUFFER
005176' 042702  177770                  BIC     #177770,R2      ;LEAVE LSD
005202' 116104  000007                  MOVB    VAM(R1),R4      ;GET LAST BUFFER MASK.
005206' 042704  177770                  BIC     #177770,R4      ;LEAVE LSD
005212' 110263  000001                  MOVB    R2,1(R3)        ;UPDATE THE USER STATUS WORD
005216' 120204                          CMPB    R2,R4           ;LAST BUFFER ?
005220' 003402                          BLE     OUTCON          ;NO-BRANCH
005222' 052713  040000                  BIS     #40000,(R3)     ;SET STOP JOB BIT
005226' 042777  000200  173000  OUTCON: BIC     #BIT7,@LPCD     ;CLEAR "READY OUT" - WHICH REENABLES THE LPA-11

005234'                         OUPEX:
005234' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```

```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #0
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB0U WILL BE ALTERED BY THE LPA-11

005240' 000000          JOB0:   .WORD   0               ;/MODE INFORMATION.
005242' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005244' 005332'                 .WORD   0+JOB0U ;WORD 4 ;/USW (USER STATUS WORD)
005246'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005247'     000                 .BYTE   0       ;BYTE 7 ;/VRM (VALID BUFFER MASK)
005250' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005252'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005253'     000                 .BYTE   0               ;/UNUSED
005254' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005256'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005257'     000                 .BYTE   0               ;/UNUSED
005260' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005262'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005263'     000                 .BYTE   0               ;/UNUSED
005264' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005266'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005267'     000                 .BYTE   0               ;/UNUSED
005270' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005272'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005273'     000                 .BYTE   0               ;/UNUSED
005274' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005276'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005277'     000                 .BYTE   0               ;/UNUSED
005300' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005302'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005303'     000                 .BYTE   0               ;/UNUSED
005304' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005306'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005307'     000                 .BYTE   0               ;/UNUSED
005310' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
005312'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
005313'     000                 .BYTE   0               ;/UNUSED.
005314' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005316'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005317'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005320' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005322' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
005324'     000                 .BYTE   0               ;/STWD
005325'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
005326' 000000                  .WORD   0               ;/ST MSK
005330' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REG. TABLE


005332' 000000          JOB0U: .WORD    0               ;/USW USER STATUS WORD
```



```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #1
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB1U WILL BE ALTERED BY THE LPA-11


005334' 000000          JOB1:   .WORD   0               ;/MODE INFORMATION.
005336' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005340' 005426'                 .WORD   0+JOB1U ;WORD 4 ;/USW (USER STATUS WORD)
005342'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005343'     000                 .BYTE   0       ;BYTE 7 ;/VRM (VALID BUFFER MASK)
005344' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005346'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005347'     000                 .BYTE   0               ;/UNUSED
005350' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005352'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005353'     000                 .BYTE   0               ;/UNUSED
005354' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005356'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005357'     000                 .BYTE   0               ;/UNUSED
005360' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005362'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005363'     000                 .BYTE   0               ;/UNUSED
005364' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005366'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005367'     000                 .BYTE   0               ;/UNUSED
005370' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005372'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005373'     000                 .BYTE   0               ;/UNUSED
005374' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005376'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005377'     000                 .BYTE   0               ;/UNUSED
005400' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005402'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005403'     000                 .BYTE   0               ;/UNUSED
005404' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
005406'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
005407'     000                 .BYTE   0               ;/UNUSED.
005410' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005412'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005413'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005414' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005416' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
005420'     000                 .BYTE   0               ;/STWD
005421'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
005422' 000000                  .WORD   0               ;/ST MSK
005424' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REG. TABLE


005426' 000000          JOB1U: .WORD    0               ;/USW USER STATUS WORD
```

```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #2
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB2U WILL BE ALTERED BY THE LPA-11

005430' 000000          JOB2:   .WORD   0               ;/MODE INFORMATION.
005432' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005434' 005522'                 .WORD   0+JOB2U ;WORD 4 ;/USW (USER STATUS WORD)
005436'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005437'     000                 .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
005440' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005442'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005443'     000                 .BYTE   0               ;/UNUSED
005444' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005446'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005447'     000                 .BYTE   0               ;/UNUSED
005450' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005452'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005453'     000                 .BYTE   0               ;/UNUSED
005454' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005456'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005457'     000                 .BYTE   0               ;/UNUSED
005460' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005462'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005463'     000                 .BYTE   0               ;/UNUSED
005464' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005466'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005467'     000                 .BYTE   0               ;/UNUSED
005470' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005472'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005473'     000                 .BYTE   0               ;/UNUSED
005474' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005476'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005477'     000                 .BYTE   0               ;/UNUSED
005500' 000000                  .WORD   0       ;WORD50 ;/RCLP (ADDR OF LIST OF RANDOM CHAN)
005502'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
005503'     000                 .BYTE   0               ;/UNUSED.
005504' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005506'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005507'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005510' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005512' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
005514'     000                 .BYTE   0               ;/STWD
005515'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
005516' 000000                  .WORD   0               ;/ST MSK
005520' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REG. TABLE

005522' 000000          JOB2U: .WORD    0               ;/USW USER STATUS WORD
```



```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #3
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB3U WILL BE ALTERED BY THE LPA-11

005524' 000000          JOB3:   .WORD   0               ;/MODE INFORMATION.
005526' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005530' 005616'                 .WORD   0+JOB3U ;WORD 4 ;/USW (USER STATUS WORD)
005532'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005533'     000                 .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
005534' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005536'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005537'     000                 .BYTE   0               ;/UNUSED
005540' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005542'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005543'     000                 .BYTE   0               ;/UNUSED
005544' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005546'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005547'     000                 .BYTE   0               ;/UNUSED
005550' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005552'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005553'     000                 .BYTE   0               ;/UNUSED
005554' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005556'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005557'     000                 .BYTE   0               ;/UNUSED
005560' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005562'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005563'     000                 .BYTE   0               ;/UNUSED
005564' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005566'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005567'     000                 .BYTE   0               ;/UNUSED
005570' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005572'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
005573'     000                 .BYTE   0               ;/UNUSED
005574' 000000                  .WORD   0       ;WORD50 ;/RCLP (ADDR OF LIST OF RANDOM CHAN)
005576'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
005577'     000                 .BYTE   0               ;/UNUSED.
005600' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005602'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005603'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005604' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005606' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
005610'     000                 .BYTE   0               ;/STWD
005611'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
005612' 000000                  .WORD   0               ;/ST MSK
005614' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REG. TABLE

005616' 000000          JOB3U: .WORD    0               ;/USW USER STATUS WORD
```

```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #4
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB4U WILL BE ALTERED BY THE LPA-11

005620' 000000          JOB4:   .WORD   0               ;/MODE INFORMATION.
005622' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005624' 005712'                 .WORD   0+JOB4U ;WORD 4 ;/USW (USER STATUS WORD)
005626'    000                  .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005627'    000                  .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
005630' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005632'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005633'    000                  .BYTE   0               ;/UNUSED
005634' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005636'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005637'    000                  .BYTE   0               ;/UNUSED
005640' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005642'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005643'    000                  .BYTE   0               ;/UNUSED
005644' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005646'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005647'    000                  .BYTE   0               ;/UNUSED
005650' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005652'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005653'    000                  .BYTE   0               ;/UNUSED
005654' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005656'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005657'    000                  .BYTE   0               ;/UNUSED
005660' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005662'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005663'    000                  .BYTE   0               ;/UNUSED
005664' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005666'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005667'    000                  .BYTE   0               ;/UNUSED
005670' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
005672'    000                  .BYTE   0               ;/RCL EXTENDED ADR. BITS
005673'    000                  .BYTE   0               ;/UNUSED.
005674' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005676'    000                  .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005677'    000                  .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005700' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005702' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
005704'    000                  .BYTE   0               ;/STWD
005705'    000                  .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
005706' 000000                  .WORD   0               ;/SI MSK
005710' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REQ. TABLE

005712' 000000          JOB4U: .WORD    0               ;/USW USER STATUS WORD
```



```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #5
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB5U WILL BE ALTERED BY THE LPA-11

005714' 000000          JOB5:   .WORD   0               ;/MODE INFORMATION.
005716' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
005720' 006006'                 .WORD   0+JOB5U ;WORD 4 ;/USW (USER STATUS WORD)
005722'    000                  .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
005723'    000                  .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
005724' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
005726'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005727'    000                  .BYTE   0               ;/UNUSED
005730' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
005732'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005733'    000                  .BYTE   0               ;/UNUSED
005734' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
005736'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005737'    000                  .BYTE   0               ;/UNUSED
005740' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
005742'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005743'    000                  .BYTE   0               ;/UNUSED
005744' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
005746'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005747'    000                  .BYTE   0               ;/UNUSED
005750' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
005752'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005753'    000                  .BYTE   0               ;/UNUSED
005754' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
005756'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005757'    000                  .BYTE   0               ;/UNUSED
005760' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
005762'    000                  .BYTE   0               ;/EXTENDED ADDRESS BITS
005763'    000                  .BYTE   0               ;/UNUSED
005764' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
005766'    000                  .BYTE   0               ;/RCL EXTENDED ADR. BITS
005767'    000                  .BYTE   0               ;/UNUSED.
005770' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
005772'    000                  .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
005773'    000                  .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
005774' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
005776' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
006000'    000                  .BYTE   0               ;/STWD
006001'    000                  .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
006002' 000000                  .WORD   0               ;/SI MSK
006004' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                ;END REQ. TABLE

006006' 000000          JOB5U: .WORD    0               ;/USW USER STATUS WORD
```

```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #6
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB6U WILL BE ALTERED BY THE LPA-11
006010' 000000          JOB6:   .WORD   0               ;/MODE INFORMATION.
006012' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
006014' 006102'                 .WORD   0+JOB6U ;WORD 4 ;/USW (USER STATUS WORD)
006016'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
006017'     000                 .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
006020' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
006022'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006023'     000                 .BYTE   0               ;/UNUSED
006024' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
006026'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006027'     000                 .BYTE   0               ;/UNUSED
006030' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
006032'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006033'     000                 .BYTE   0               ;/UNUSED
006034' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
006036'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006037'     000                 .BYTE   0               ;/UNUSED
006040' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
006042'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006043'     000                 .BYTE   0               ;/UNUSED
006044' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
006046'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006047'     000                 .BYTE   0               ;/UNUSED
006050' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
006052'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006053'     000                 .BYTE   0               ;/UNUSED
006054' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
006056'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006057'     000                 .BYTE   0               ;/UNUSED
006060' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
006062'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
006063'     000                 .BYTE   0               ;/UNUSED.
006064' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
006066'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
006067'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
006070' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
006072' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
006074'     000                 .BYTE   0               ;/STWD
006075'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
006076' 000000                  .WORD   0               ;/ST MSK
006100' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                        ;END REG. TABLE

006102' 000000          JOB6U: .WORD    0               ;/USW USER STATUS WORD
```



```
                                        ;**     REQUEST DESCRIPTOR ARRAY FOR JOB #7
                                        ;**
                                        ;**     THIS TABLE WILL BE ALTERED BY THIS PROGRAM
                                        ;**     LOCATION JOB7U WILL BE ALTERED BY THE LPA-11
006104' 000000          JOB7:   .WORD   0               ;/MODE INFORMATION.
006106' 000000                  .WORD   0       ;WORD 2 ;/WORD COUNT
006110' 006176'                 .WORD   0+JOB7U ;WORD 4 ;/USW (USER STATUS WORD)
006112'     000                 .BYTE   0       ;BYTE 6 ;/USW (EXTENDED ADDR. BITS)
006113'     000                 .BYTE   0       ;BYTE 7 ;/VBM (VALID BUFFER MASK)
006114' 000000                  .WORD   0       ;WORD10 ;/BUFFER ADDRESS #0
006116'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006117'     000                 .BYTE   0               ;/UNUSED
006120' 000000                  .WORD   0       ;WORD14 ;/BUFFER ADDRESS #1
006122'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006123'     000                 .BYTE   0               ;/UNUSED
006124' 000000                  .WORD   0       ;WORD20 ;/BUFFER ADDRESS #2
006126'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006127'     000                 .BYTE   0               ;/UNUSED
006130' 000000                  .WORD   0       ;WORD24 ;/BUFFER ADDRESS #3
006132'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006133'     000                 .BYTE   0               ;/UNUSED
006134' 000000                  .WORD   0       ;WORD30 ;/BUFFER ADDRESS #4
006136'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006137'     000                 .BYTE   0               ;/UNUSED
006140' 000000                  .WORD   0       ;WORD34 ;/BUFFER ADDRESS #5
006142'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006143'     000                 .BYTE   0               ;/UNUSED
006144' 000000                  .WORD   0       ;WORD40 ;/BUFFER ADDRESS #6
006146'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006147'     000                 .BYTE   0               ;/UNUSED
006150' 000000                  .WORD   0       ;WORD44 ;/BUFFER ADDRESS #7
006152'     000                 .BYTE   0               ;/EXTENDED ADDRESS BITS
006153'     000                 .BYTE   0               ;/UNUSED
006154' 000000                  .WORD   0       ;WORD50 ;/RCLR (ADDR OF LIST OF RANDOM CHAN)
006156'     000                 .BYTE   0               ;/RCL EXTENDED ADR. BITS
006157'     000                 .BYTE   0               ;/UNUSED.
006160' 000000                  .WORD   0       ;WORD54 ;DELAY WORD (COUNTER AFTER "START COMMAND")
006162'     000                 .BYTE   0       ;BYTE56 ;/START CHANNEL NUMBER
006163'     000                 .BYTE   0       ;BYTE57 ;/CHANNEL NUMBER INCREMENT
006164' 000000                  .WORD   0       ;WORD60 ;/NUMBER OF SAMPLES IN A SAMPLE SEQUENCE
006166' 000000                  .WORD   0       ;WORD62 ;/SAMPLE RATE (DWELL VALUE)
006170'     000                 .BYTE   0               ;/STWD
006171'     000                 .BYTE   0               ;/EMWD EVENT MARK DIGITAL INT WD#
006172' 000000                  .WORD   0               ;/ST MSK
006174' 000000                  .WORD   0               ;/EM MSK EVENT MARK DIGITAL INPUT MASK.
                                        ;END REG. TABLE

006176' 000000          JOB7U: .WORD    0               ;/USW USER STATUS WORD
```

```
                                WARNMSG:        WRNMSG
                                                -1
006202' 177777
006204' 006412  020040  020040  WRNMSG: .ASCII  <12><15>#     WARNING: #
006212' 040527  047122  047111
006220' 035107     040
006223'    012  052015  044510          .ASCII  <12><15>#THIS MODULE CAN'T RUN UNLESS SR1 HAS BEEN SETUP#
006230' 020123  047515  052504
006236' 042514  041440  047101
006244' 052047  051040  047125
006252' 052440  046116  051505
006260' 020123  051123  020061
006266' 040510  020123  042502
006274' 047105  051440  052105
006302' 050125
006304' 006412  047524  051040          .ASCIZ  <12><15>#TO REFLECT THE TRUE LPA11 I/O BUS CONFIGURATION#
006312' 043105  042514  052103
006320' 052040  042510  052040
006326' 052522  020105  050114
006334' 030501  020061  027511
006342' 020117  052502  020123
006350' 047503  043116  043511
006356' 051125  052101  047511
006364' 000116

                                        .EVEN

                                        ;**
                                        ;** REQUEST DESCRIPTOR ARRAY FOR THE LPA-11 CLOCK
                                        ;** LOCATION "KWTPRE" WILL BE ALTERED, IF "DEDICATED
                                        ;**  MICRO-CODE" IS SELECTED, WITH RESPECT TO THE VALUE OF "SR2".

006366' 000011          KWT:    11              ;START CLOCK CODE
006370' 000503                  503             ;CLOCK STATUS AND MODE VALUE
006372' 000000          KWTPRE: 0               ;CLOCK PRESET BUFFER VALUE

                        ;SUBROUTINE TO STOP THE LPA-11K "CLEAN AND NICE"
006374' 016767  171634  171476  RESET0: MOV     LPC0,CSRA       ;LOAD LPA CONTROL ADDRESS
006402' 017767  171626  171472          MOV     @LPC0,ACSR      ;GET CONTENTS OF OUT REGISTER
006410' 017767  171614  171466          MOV     @LPC1,ASTAT     ;GET CONTENTS OF IN REGISTER
006416' 042777  000100  171610          BIC     #BIT6,@LPC0     ;REMOVE INTR. ENABLE
006424' 042777  000100  171576          BIC     #BIT6,@LPC1     ; FROM THE LPA-11K
006432' 052777  040000  171570          BIS     #BIT14,@KMAD0   ;SET "LPA-11 RESET" BIT
006440' 012700  000002                  MOV     #2,R0           ;LOAD DELAY COUNTER
006444'                 1$:
006444' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
006450' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
006454' 005300                          DEC     R0              ;DELAY A PERIOD
006456' 001372                          BNE     1$              ; OF TIME
006460' 042777  040000  171542          BIC     #BIT14,@KMAD0   ;REMOVE "RESET" BIT
006466' 000207                          RTS     PC              ;EXIT SUBROUTINE
```



```
006470' 000000          DEVLST: .WORD   0               ;MODE WORD FOR START.
006472' 000012                  .BLKW   10.             ;TEN ADDRESSES ON START.
006516'                 DMDT:
                                .SBTTL  DMDT -- DEDICATED MODE DISPATCH TABLE
                        ;       .IDENT  /LPA.03/
                        ;
                        ;
                        ;   COPYRIGHT (C) 1976, 1980, 1981  DIGITAL EQUIPMENT CORP.,   MAYNARD,   MASS.
                        ;
                        ; THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE
                        ; ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION
                        ; OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM,
                        ; EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.
                        ;
                        ; THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT
                        ; NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL
                        ; EQUIPMENT CORPORATION.
                        ;
                        ; DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF
                        ; ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
                        ;
                        ;
                        ; CHARLES A. SAMUELSON
                        ; FEBRUARY 4, 1977
                        ;
                        ; +
                        ; DMDT -- DEDICATED MODE DISPATCH TABLE
                        ; TABLE FOR LPA11 MICRO PROCESSOR DEDICATED MODE SAMPLING
                        ; -
                        ;
                        ; DEFINED VALUES
                        ;
000022                  DMDSIZ=18.              ;LENGTH OF DMDT BUFFER IN BYTES
000100                  SDT=100                 ;SLAVE DISPATCH TABLE START ADDRESS
000000                  CAINC=0                 ;CHANNEL ADDRESS INCREMENT VALUE
000000                  AD1SRL=0                ;ADC #1 STATUS REGISTER ADDRESS LOW BYTE
000100                  SEX=100                 ;SELECT EXTERNAL CLOCK START AND INTERRUPT ENABLE
000035                  RONPR=35                ;REQUEST OUTPUT NPR IN MICRO-PROCESSOR
000000                  CLR=0                   ;CLEAR AD STATUS REGISTER
000235                  RONPRL=235              ;REQUEST OUTPUT NPR LOW BYTE IN MICRO-PROCESSOR
000002                  AD1DRL=2                ;ADC #1 DATA REGISTER ADDRESS LOW BYTE
000015                  RINPR=15                ;REQUEST INPUT NPR
000040                  SCS=40                  ;SELECT CLOCK OVERFLOW START FOR ADC'S
000040                  AD2SRL=40               ;ADC #2 STATUS REGISTER ADDRESS LOW BYTE
000042                  AD2DRL=42               ;ADC #2 DATA REGISTER ADDRESS LOW BYTE
000001                  AD1SRH=1                ;ADC #1 STATUS REGSITER HIGH BYTE ADDRESS
000041                  AD2SRH=41               ;ADC #2 STATUS REGISTER HIGH BYTE ADDRESS
000020                  SEN=20                  ;SELECT EXTERNAL START, NO INTERRUPT ENABLE
                        ;
                        ;
006516'                 D.OES:                  ;ONE ADC, EXTERNAL TRIGGER, SINGLE CHANNEL
006516'    162     000     100          .BYTE   DMDSIZ+<3*40>,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
006521'    035     002     015
006524'    000     235     100
          006541'                       .=D.OES+23
006541'                 D.OEQ:                  ;ONE ADC, EXTERNAL TRIGGER, SEQUENTIAL CHANNEL
```

```
1592  006541'     162      000      000          .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
1593  006544'     100      035      002
1594  006547'     015      000      235
1595  006552'     100
1596  006553'     000                           .BYTE   CAINC
1597            006564'                  .=D.DEQ+23
1598  006564'                            D.OCS:                ;ONE ADC, CLOCK TRIGGER, SINGLE CHANNEL
1599  006564'     162      000      040          .BYTE   DMDSIZ+<3*40>,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
1600  006567'     035      002      015
1601  006572'     000      235      100
1602            006607'                  .=D.OCS+23
1603  006607'                            D.OCQ:                ;ONE ADC, CLOCK TRIGGER, SEQUENTIAL CHANNEL
1604  006607'     162      000      000          .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT
1605  006612'     040      035      002
1606  006615'     015      000      235
1607  006620'     100
1608  006621'     000                           .BYTE   CAINC
1609            006632'                  .=D.OCQ+23
1610  006632'                            D.TES:                ;TWO ADC, EXTERNAL TRIGGER, SINGLE CHANNEL
1611  006632'     162      000      100          .BYTE   DMDSIZ+<3*40>,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL,SDT+10
1612  006635'     035      002      015
1613  006640'     000      235      110
1614  006643'     040      100      035          .BYTE   AD2SRL,SEX,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
1615  006646'     042      015      000
1616  006651'     235      100
1617            006655'                  .=D.TES+23
1618  006655'                            D.TEQ:                ;TWO ADC, EXTERNAL TRIGGER, SEQUENTIAL CHANNEL
1619  006655'     162      000      000          .BYTE   DMDSIZ+<3*40>,CAINC,AD1SRL,SEX,RONPR,AD1DRL,RINPR,CLR,RONPRL
1620  006660'     100      035      002
1621  006663'     015      000      235
1622  006666'     111      000      040          .BYTE   SDT+11,CAINC,AD2SRL,SEX,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
1623  006671'     100      035      042
1624  006674'     015      000      235
1625  006677'     100
1626            006700'                  .=D.TEQ+23
1627  006700'                            D.TCS:                ;TWO ADC, CLOCK TRIGGER, SINGLE CHANNEL
1628  006700'     262      000      040          .BYTE   DMDSIZ+<5*40>,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL
1629  006703'     035      002      015
1630  006706'     000      235
1631  006710'     110      040      040          .BYTE   SDT+10,AD2SRL,SCS,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
1632  006713'     035      042      015
1633  006716'     000      235      100
1634            006723'                  .=D.TCS+23
1635  006723'                            D.TCQ:                ;TWO ADC, CLOCK TRIGGER, SEQUENTIAL CHANNEL
1636  006723'     262      000      000          .BYTE   DMDSIZ+<5*40>,CAINC,AD1SRL,SCS,RONPR,AD1DRL,RINPR,CLR,RONPRL
1637  006726'     040      035      002
1638  006731'     015      000      235
1639  006734'     111      000      040          .BYTE   SDT+11,CAINC,AD2SRL,SCS,RONPR,AD2DRL,RINPR,CLR,RONPRL,SDT
1640  006737'     040      035      042
1641  006742'     015      000      235
1642  006745'     100
1643            006746'                  .=D.TCQ+23
1644                                     ;
1645                                     ; PARALLEL MODE TABLE
1646                                     ;
1647  006746'                            D.TESP:               ;TWO ADC, EXTERNAL TRIGGER, SINGLE, PARALLELS
```



```
1648  006746'     022      100      000          .BYTE   DMDSIZ+<0*40>,SEX,AD1SRL,RONPR,AD2SRL,SEX,RONPR,AD1DRL,RINPR
1649  006751'     035      040      020
1650  006754'     035      002      015
1651  006757'     042      015      001          .BYTE   AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
1652  006762'     235      041      235
1653  006765'     106
1654            006771'                  .=D.TESP+23
1655  006771'                            D.TEQP:               ;TWO ADC, EXTERNAL TRIGGER, SEQUENTIAL, PARALLEL
1656  006771'     022      100      000          .BYTE   DMDSIZ+<0*40>,SEX,AD1SRL,RONPR,AD2SRL,SEX,RONPR,CAINC,AD1DRL
1657  006774'     035      040      020
1658  006777'     035      000      002
1659  007002'     015      042      015          .BYTE   RINPR,AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
1660  007005'     001      235      041
1661  007010'     235      106
1662            007014'                  .=D.TEQP+23
1663  007014'                            D.TCSP:               ;TWO ADC, CLOCK TRIGGER, SINGLE, PARALLEL
1664  007014'     222      040      000          .BYTE   DMDSIZ+<4*40>,SCS,AD1SRL,RONPR,AD2SRL,SCS,RONPR,AD1DRL,RINPR
1665  007017'     035      040      040
1666  007022'     035      002      015
1667  007025'     042      015      001          .BYTE   AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
1668  007030'     235      041      235
1669  007033'     106
1670            007037'                  .=D.TCSP+23
1671  007037'                            D.TCQP:               ;TWO ADC, CLOCK TRIGGER, SEQUENTIAL, PARALLEL
1672  007037'     222      040      000          .BYTE   DMDSIZ+<4*40>,SCS,AD1SRL,RONPR,AD2SRL,SCS,RONPR,CAINC,AD1DRL
1673  007042'     035      040      040
1674  007045'     035      000      002
1675  007050'     015      042      015          .BYTE   RINPR,AD2DRL,RINPR,AD1SRH,RONPRL,AD2SRH,RONPRL,SDT+6
1676  007053'     001      235      041
1677  007056'     235      106
1678
```

```
                                ;THIS IS THE "TO DO" QUEUE FOR THE "INPUT" INTERRUPTS

007060' 000050          TODOQ:  .BLKW 40.
007200' 000000          TODOC:  .WORD 0

                                ;THE FOLLOWING TABLE IS USED BY "ELOOP" TO INDEX THE JOB BASED
                                ;ON THE INFO THE LPA RETURNS AS ASSIGNED JOB NUMBERS.
                                ;"ELOOP" INDEXES THIS TABLE WITH THE JOB NUMBER TO STORE THE RDA
                                ;ADDR.  WHEN CODE=0, ANY OTHER TIME TABLE IS INDEXED TO GET
                                ;THE RDA ADDR. OF A PARTICULAR JOB.

007202' 000000          TABLE:  .WORD   0
007204' 000000                  .WORD   0
007206' 000000                  .WORD   0
007210' 000000                  .WORD   0
007212' 000000                  .WORD   0
007214' 000000                  .WORD   0
007216' 000000                  .WORD   0
007220' 000000                  .WORD   0
007222' 000000                  .WORD   0
                                        ;NOTE THIS AREA FROM TODOQ: TO DONEC: IS
                                        ;CLEARED EACH TIME "ELOOP" IS ENTERED.
007224' 000040                  .BLKW   32.
007324' 000000          DONEC:  .WORD 0

007326'                 BUFFER:
007326' 000401          BUFF0:  .BLKW   257.    ;1ST A TO D'S BUFFER <MULTI-USER MICRO-CODE>
010330' 000401          BUFF1:  .BLKW   257.    ;2ND A TO D'S BUFFER <MULTI-USER MICRO-CODE>
011332' 000401                  .BLKW   257.
012334' 000401          BUFF2:  .BLKW   257.    ;DIGITAL I/O BUFFER <MULTI-USER MICRO-CODE>
013336' 000401          BUFF3:  .BLKW   257.    ;D TO A BUFFER <MULTI-USER MICRO-CODE>
014340' 000401                  .BLKW   257.
                                .EVEN


                        ;THE 2 DIFFERENT MICRO-CODE'S THAT ARE USED BY THIS MODULE RESIDE
                        ;       HERE TO LOCATION "ALDONE"
                        ;MULTI-USER MICRO-CODE STARTS AT LOCATION
015342' 015346'                 MMAST
                        ;DEDICATED A TO D MICRO-CODE STARTS AT LOCATION
015344' 021352'                 DMAST
```



```
                                .SBTTL  LISTING OF MULTI-USER MICROCODE
015346'                 MMAST:
015346' 000400  071220  065222          .WORD     400, 71220, 65222, 63224, 63226,   777, 63223, 63230
015354' 063224  063226  000777
015362' 063223  063230
015366' 063231  000407  063220          .WORD    63231,   407, 63220, 16777, 63160,101013,   600, 63222
015374' 016777  063160  101013
015402' 000600  063222
015406' 020640  061620  103024          .WORD    20640, 61620,103024,100431, 63170,101020, 63171,101020
015414' 100431  063170  101020
015422' 063171  101020
015426' 104606  114717  104607          .WORD   104606,114717,104607,   452, 23240, 60360,115660,104610
015434' 000452  023240  060360
015442' 115660  104610
015446' 070605  063224  024360          .WORD    70605, 63224, 24360, 53221,101451, 64201, 43637,106000
015454' 053221  101451  064201
015462' 043637  106000
015466' 113472  063165  105373          .WORD   113472, 63165,105373,   407, 63225, 23250, 63530,107612
015474' 000407  063225  023250
015502' 063530  107612
015506' 103062  104613  060530          .WORD   103062,104613, 60530,107211, 60603,101467,104565,120400
015514' 107211  060603  101467
015522' 104565  120400
01552e' 102163  123451  107773          .WORD   102163,123451,107773,120620,107773, 60522,103550,123013
015534' 120620  107773  060522
015542' 103550  123013
015546' 060602  103506  102143          .WORD    60602,103506,102143, 60613,103533,104773,   466, 62224
015554' 060613  103533  104773
015562' 000466  062224
015566' 000761  062225  000417          .WORD      761, 62225,   417, 61230,120600,102114,123014,   600
015574' 061230  120600  102114
015602' 123014  000600
015606' 060714  061220  000414          .WORD    60714, 61220,   414, 63220,   417, 61230, 61210,   577
015614' 063220  000417  061230
015622' 061210  000577
015626' 063262  120620  106202          .WORD    63262,120620,106202, 60533,103536,100545,   700, 61231
015634' 060533  103536  100545
015642' 000700  061231
015646' 000401  063302  104773          .WORD      401, 63302,104773, 60533,103540,   576, 63262,104773
015654' 060533  103540  000576
015662' 063262  104773
015666' 060611  103553  100560          .WORD    60611,103553,100560, 60531,103556,100477,   600, 61231
015674' 060531  103556  100477
015702' 000600  061231
015706' 000677  063262  100471          .WORD      677, 63262,100471,122104,122125,   414,123000, 63260
015714' 122104  122125  000414
015722' 123000  063260
015726' 000403  060700  061230          .WORD      403, 60700, 61230,120600,102173,   576,123011, 60671
015734' 120600  102173  000576
015742' 123011  060671
015746' 061220  000403  060700          .WORD    61220,   403, 60700, 61230, 61210, 60611,103610,104545
015754' 061230  061210  060611
015762' 103610  104545
015766' 020500  106153  120620          .WORD    20500,106153,120620,106147, 27364, 23000,   407, 63260
015774' 106147  027364  023000
016002' 000407  063260
```

```
1776  016006' 000622  160400  104735          .WORD    622,160400,104735,100710,100722,114643,104550,104550
1777  016014' 100710  100722  114643
1778  016022' 104550  104550
1779  016026' 104550  104550  063237          .WORD  104550,104550, 63237,   453, 70001, 42224, 43230, 63070
1780  016034' 000453  070001  042224
1781  016042' 043230  063070
1782  016046' 076470  101274  056225          .WORD   76470,101274, 56225, 41230, 20640,107476,120600,102246
1783  016054' 041230  020640  107476
1784  016062' 120600  102246
1785  016066' 120620  102306  000440          .WORD  120620,102306,   440, 60704, 62230, 20640,102255, 22010
1786  016074' 060704  062230  020640
1787  016102' 102255  022010
1788  016106' 020420  103663  170617          .WORD   20420,103663,170617,   450, 70001, 57230, 57231, 57232
1789  016114' 000450  070001  057230
1790  016122' 057231  057232
1791  016126' 076610  076611  062472          .WORD   76610, 76611, 62472,170617, 42225, 43230, 76470,101301
1792  016134' 170617  042225  043230
1793  016142' 076470  101301
1794  016146' 100643  041230  043230          .WORD  100643, 41230, 43230,   404, 62410,100644, 22270,104503
1795  016154' 000404  062410  100644
1796  016162' 022270  104503
1797  016166' 060602  103313  104556          .WORD   60602,103313,104556, 10012, 20400,103317, 10010, 40620
1798  016174' 010012  020400  103317
1799  016202' 010010  040620
1800  016206' 106151  100743  060602          .WORD  106151,100743, 60602,103325,104556, 23010, 63530,103331
1801  016214' 103325  104556  023010
1802  016222' 063530  103331
1803  016226' 104557  107550  060530          .WORD  104557,107550, 60530,107550, 20420, 61620,102767, 20420
1804  016234' 107550  020420  061620
1805  016242' 102767  020420
1806  016246' 102342  100743  106550          .WORD  102342,100743,106550, 10407, 63224,   703, 63221,100752
1807  016254' 010407  063224  000703
1808  016262' 063221  100752
1809  016266' 000702  063221  040620          .WORD     702, 63221, 40620,101764, 73164,105562,   700, 63001
1810  016274' 101764  073164  105562
1811  016302' 000700  063001
1812  016306' 000403  060364  101750          .WORD     403, 60364,101750,100752, 72601, 64201,104750, 20400
1813  016314' 100752  072601  064201
1814  016322' 104750  020400
1815  016326' 103372  104550  020420          .WORD  103372,104550, 20420,102342,104550,     0,     0,     0
1816  016334' 102342  104550  000000
1817  016342' 000600  000000
1818  016346' 120440  103451  134620          .WORD  120440,103451,134620,103451, 43626,107432, 60566,105443
1819  016354' 103451  043626  107432
1820  016362' 060566  105443
1821  016366' 000401  060366  105422          .WORD     401, 60366,105422,   456, 70001, 43234, 62574,105022
1822  016374' 000456  070001  043234
1823  016402' 062574  105022
1824  016406' 003006  104505  070201          .WORD    3006,104505, 70201, 43234,   576, 76674, 22760,   431
1825  016414' 043234  000576  076674
1826  016422' 022760  000431
1827  016426' 104646  104772  070204          .WORD  104646,104772, 70204,  4000,  2777, 64201, 63537,107443
1828  016434' 004000  002777  064201
1829  016442' 063537  107443
1830  016446' 060537  107450  104427          .WORD   60537,107450,104427,   600, 63302, 60537,107366,104422
1831  016454' 000600  063302  060537
```



```
1832  016462' 107366  104422
1833  016466' 000653  060366  105457          .WORD     653, 60366,105457,   654, 60366,105457,104427,   650
1834  016474' 000654  060366  105457
1835  016502' 104427  000650
1836  016506' 063226  104427  063066          .WORD  63226,104427, 63066, 63066, 63066,   724, 63006, 70201
1837  016514' 063066  063066  000724
1838  016522' 063006  070201
1839  016526' 064201  016501  062606          .WORD   64201, 16501, 62606,104766, 63066, 63066, 63066, 63066
1840  016534' 104766  063066  063066
1841  016542' 063066  063066
1842  016546' 063066  063066  063066          .WORD  63066, 63066, 63066, 63066, 63066,   640, 63306, 70201
1843  016554' 063066  063066  000640
1844  016562' 063306  070201
1845  016566' 054620  107366  106114          .WORD  54620,107366,106114,104516, 40620,107766, 62606, 20640
1846  016574' 104516  040620  107766
1847  016602' 062606  020640
1848  016606' 107117  000420  060704          .WORD  107117,   420, 60704, 62231, 70201, 43230,   420, 62710
1849  016614' 062231  070201  043230
1850  016622' 000420  062710
1851  016626' 104766  063066  063066          .WORD  104766, 63066, 63066, 70201, 43631,106143,107372,   401
1852  016634' 070201  043631  106143
1853  016642' 107372  000401
1854  016646' 076711  062606  104772          .WORD   76711, 62606,104772, 23366,104504, 63066,104562, 63066
1855  016654' 023366  104504  063066
1856  016662' 104562  063066
1857  016666' 063066  063066  063066          .WORD  63066, 63066, 63066, 63066, 63066, 63066, 63066, 63066
1858  016674' 063066  063066  063066
1859  016702' 063066  063066
1860  016706' 063066  063126  000700          .WORD  63066, 63126,   700, 60706, 63223, 63226,120440,107766
1861  016714' 060706  063223  063226
1862  016722' 120440  107766
1863  016726' 120620  107766  023263          .WORD  120620,107766, 23263,   575,104646, 60606,106366,   600
1864  016734' 000575  104646  060606
1865  016742' 106366  000600
1866  016746' 063302  104766  063066          .WORD  63302,104766, 63066, 63066, 63066, 63066, 63066, 63066
1867  016754' 063066  063066  063066
1868  016762' 063066  063066
1869  016766' 063066  063066  063066          .WORD  63066, 63066, 63066, 61226, 61207,   740, 63306,   400
1870  016774' 061226  061207  000740
1871  017002' 063306  000400
1872  017006' 063236  120620  107621          .WORD  63236,120620,107621,123451,107633, 60616,105621,   631
1873  017014' 123451  107633  060616
1874  017022' 105621  000631
1875  017026' 104646  063176  104621          .WORD  104646, 63176,104621, 60522,107636,104623, 60531,107641
1876  017034' 060522  107636  104623
1877  017042' 060531  107641
1878  017046' 104623  000600  061231          .WORD  104623,   600, 61231,   677, 63262,104621, 63237, 60606
1879  017054' 000677  063262  104621
1880  017062' 063237  060606
1881  017066' 061223  000466  062224          .WORD  61223,   466, 62224,   761, 62225,   417, 61230,120600
1882  017074' 000761  062225  000417
1883  017102' 061230  120600
1884  017106' 106257  123052  000500          .WORD  106257,123052,   500, 63272,   600, 60704, 60712, 61222
1885  017114' 063272  000600  060704
1886  017122' 060712  061222
1887  017126' 000414  063220  000417          .WORD     414, 63220,   417, 61230, 61210, 60532,107702,   500
```

```
1888  017134' 061230  061210  060532
1889  017142' 107702  000500
1890  017146' 063302  164617  000600          .WORD    63302,164617,   600, 61231,   677, 63262,164617, 63236
1891  017154' 061231  000677  063262
1892  017162' 164617  063236
1893  017166' 000404  070001  042224          .WORD      404, 70001, 42224, 43230, 63070, 76470,105324, 56225
1894  017174' 043230  063070  076470
1895  017202' 105324  056225
1896  017206' 057220  063060  061210          .WORD    57220, 63060, 61210,170616, 42225, 43230, 76470,105331
1897  017214' 170616  042225  043230
1898  017222' 076470  105331
1899  017226' 104720  043220  000404          .WORD   104720, 43220,   404, 76400,104721, 60602,107155, 23030
1900  017234' 076400  104721  060602
1901  017242' 107155  023030
1902  017246' 000405  060370  105744          .WORD      405, 60370,105744,104552, 30360,  2402, 53221, 64201
1903  017254' 104552  030360  002402
1904  017262' 053221  064201
1905  017266' 016600  036760  036420          .WORD    16600, 36760, 36420, 36400, 36500, 36520,123210,   414
1906  017274' 036400  036500  036520
1907  017302' 123210  000414
1908  017306' 076670  002502  000456          .WORD    76670,  2502,   456, 70401, 16401,104773,123230,   556
1909  017314' 070401  016401  104773
1910  017322' 123230  000556
1911  017326' 060670  061231  023366          .WORD    60670, 61231, 23366, 20640,116421, 23604, 63232,   417
1912  017334' 020640  116421  023604
1913  017342' 063232  000417
1914  017346' 073264  004000  053221          .WORD    73264,  4000, 53221, 64201, 60532,113455, 54620,113061
1915  017354' 064201  060532  113455
1916  017362' 054620  113061
1917  017366' 061620  107373  074612          .WORD    61620,107373, 74612,113022, 20640,112414, 60612, 61620
1918  017374' 113022  020640  112414
1919  017402' 060612  061620
1920  017406' 117045  110447  054620          .WORD   117045,110447, 54620,112027,112427,   427,100632,   431
1921  017414' 112027  112427  000427
1922  017422' 100632  000431
1923  017426' 104707  020640  107476          .WORD   104707, 20640,107476,120600,112033,   600, 60704, 62230
1924  017434' 120600  112033  000600
1925  017442' 060704  062230
1926  017446' 020640  112040  022010          .WORD    20640,112040, 22010, 20640,112043, 22030,114471, 40620
1927  017454' 020640  112043  022030
1928  017462' 114471  040620
1929  017466' 116045  116445  000454          .WORD   116045,116445,   454,100632,114445, 43233,   401, 62713
1930  017474' 100632  114445  043233
1931  017502' 000401  062713
1932  017506' 104773  060612  107373          .WORD   104773, 60612,107373, 20640,112463, 20600, 20640,112466
1933  017514' 020640  112463  020600
1934  017522' 020640  112466
1935  017526' 020600  104773  000474          .WORD    20600,104773,   474,104707, 43230,120600,112075,120620
1936  017534' 104707  043230  120600
1937  017542' 112075  120620
1938  017546' 106065  170610  002524          .WORD   106065,170610,  2524,100451, 64201, 60610,113520, 60611
1939  017554' 100451  064201  060610
1940  017562' 113520  060611
1941  017566' 112120  112520  000406          .WORD   112120,112520,   406, 63236,   516, 63237, 63176,101233
1942  017574' 063236  000516  063237
1943  017602' 063176  101233
```



```
1944  017606' 000411  070401  000533          .WORD      411, 70401,   533,114500, 60602,113131, 16541,  2406
1945  017614' 114500  060602  113131
1946  017622' 016541  002406
1947  017626' 110542  016604  000413          .WORD   110542, 16604,   413, 60401, 63236, 62476, 70220, 36400
1948  017634' 060401  063236  062476
1949  017642' 070220  036400
1950  017646' 110627  074620  043232          .WORD   110627, 74620, 43232, 63072, 72472,  4000, 20640,112146
1951  017654' 063072  072472  004000
1952  017662' 020640  112146
1953  017666' 000771  060412  061620          .WORD      771, 60412, 61620, 62230, 20640,112154, 22010, 20640
1954  017674' 062230  020640  112154
1955  017702' 022010  020640
1956  017706' 112157  022030  020640          .WORD   112157, 22030, 20640,112562, 20600,111601, 36400, 22420
1957  017714' 112562  020600  111601
1958  017722' 036400  022420
1959  017726' 000431  060372  111574          .WORD      431, 60372,111574,100451,  4620, 63302, 30360, 22660
1960  017734' 100451  004620  063302
1961  017742' 030360  022660
1962  017746' 100451  021006  021027          .WORD   100451, 21006, 21027,104463,   403, 70001, 40620,112231
1963  017754' 104463  000403  070001
1964  017762' 040620  112231
1965  017766' 000413  070401  043220          .WORD      413, 70401, 43220, 76560,111217, 43220, 62560,   407
1966  017774' 076560  111217  043220
1967  020002' 062560  000407
1968  020006' 070401  016663  043232          .WORD    70401, 16663, 43232, 72472, 20400,106064, 76620, 22420
1969  020014' 072472  020400  106064
1970  020022' 076620  022420
1971  020026' 100451  113234  000420          .WORD   100451,113234,   420,110635,   421, 63220, 20640,107476
1972  020034' 110635  000421  063220
1973  020042' 020640  107476
1974  020046' 062210  000413  070401          .WORD    62210,   413, 70401,   600, 63302, 22010, 20640,112246
1975  020054' 000600  063302  022010
1976  020062' 020640  112246
1977  020066' 022030  020640  112251          .WORD    22030, 20640,112251, 56230, 20640,112254, 42230, 70204
1978  020074' 056230  020640  112254
1979  020102' 042230  070204
1980  020106' 004000  022660  100451          .WORD     4000, 22660,100451, 16621, 43232, 63072, 72472, 23013
1981  020114' 016621  043232  063072
1982  020122' 072472  023013
1983  020126' 063133  063133  000414          .WORD    63133, 63133,   414, 62673,   451, 60401, 60372,111705
1984  020134' 062673  000451  060401
1985  020142' 060372  111705
1986  020146' 000416  060401  060372          .WORD      416, 60401, 60372,111714,100451,   407, 70401, 16731
1987  020154' 111714  100451  000407
1988  020162' 070401  016731
1989  020166' 002432  000704  063237          .WORD     2432,   704, 63237,100663,   411, 70401, 36760, 23030
1990  020174' 100663  000411  070401
1991  020202' 036760  023030
1992  020206' 000407  062670  070201          .WORD      407, 62670, 70201, 43231, 60610,103451,   402, 62711
1993  020214' 043231  060610  103451
1994  020222' 000402  062711
1995  020226' 100451  074620  043232          .WORD   100451, 74620, 43232, 63072, 72472,  4000, 36400, 22420
1996  020234' 063072  072472  004000
1997  020242' 036400  022420
1998  020246' 000447  060372  111744          .WORD      447, 60372,111744,100451, 20640,107476,   500, 60704
1999  020254' 100451  020640  107476
```

```
2000  020262' 000500  060704
2001  020266' 062230  000402  070001          .WORD   62230,   402, 70001, 64201, 43231, 56230, 20640,112356
2002  020274' 064201  043231  056230
2003  020302' 020640  112356
2004  020306' 043630  062230  020640          .WORD   43630, 62230, 20640,107476,   402, 63237, 10034,  4000
2005  020314' 107476  000402  063237
2006  020322' 010034  004000
2007  020326' 057233  057220  060573          .WORD   57233, 57220, 60573,111375, 63160, 62230, 20640,112376
2008  020334' 111375  063160  062230
2009  020342' 020640  112376
2010  020346' 062210  063577  115407          .WORD   62210, 63577,115407,116005,110770,   401,114410,   405
2011  020354' 116005  110770  000401
2012  020362' 114410  000405
2013  020366' 063234  020640  116011          .WORD   63234, 20640,116011, 56230, 63174,115011, 60477,112770
2014  020374' 056230  063174  115011
2015  020402' 060477  112770
2016  020406' 110504  061620  103040          .WORD  110504, 61620,103040, 23620,107605,   417, 60660, 73224
2017  020414' 023620  107605  000417
2018  020422' 060660  073224
2019  020426' 004000  053221  105773          .WORD    4000, 53221,105773, 64201, 60600, 61620, 61620, 61620
2020  020434' 064201  060600  061620
2021  020442' 061620  061620
2022  020446' 061620  063226  000407          .WORD   61620, 63226,   407, 63266,104474, 22202,   404, 70001
2023  020454' 063266  104474  022202
2024  020462' 000404  070001
2025  020466' 042226  043230  063070          .WORD   42226, 43230, 63070, 76470,115177, 56227, 57230,   500
2026  020474' 076470  115177  056227
2027  020502' 057230  000500
2028  020506' 123231  060671  060710          .WORD  123231, 60671, 60710, 61231, 20640,116464, 22203,   421
2029  020514' 061231  020640  116464
2030  020522' 022203  000421
2031  020526' 061230  043230  076570          .WORD   61230, 43230, 76570,115210, 43230, 76570,115210,   532
2032  020534' 115210  043230  076570
2033  020542' 115210  000532
2034  020546' 063237  057230  057231          .WORD   63237, 57230, 57231, 57232, 57233,120600,116105,123634
2035  020554' 057232  057233  120600
2036  020562' 116105  123634
2037  020566' 106101  042224  056226          .WORD  106101, 42224, 56226, 42225, 56227, 43220, 60460, 61230
2038  020574' 042225  056227  043220
2039  020602' 060460  061230
2040  020606' 000500  060674  060700          .WORD    500, 60674, 60700, 61231,120600,116124,120620,106100
2041  020614' 061231  120600  116124
2042  020622' 120620  106100
2043  020626' 000600  023036  060716          .WORD    600, 23036, 60716, 62223, 60536,117537,115215,   456
2044  020634' 062223  060536  117537
2045  020642' 115215  000456
2046  020646' 070401  002401  060610          .WORD   70401,  2401, 60610, 62222,   421, 61230, 60536,117630
2047  020654' 062222  000421  061230
2048  020662' 060536  117630
2049  020666' 000407  063676  060751          .WORD    407, 63676, 60751,107477,   420, 60401, 60416, 60416
2050  020674' 107477  000420  060401
2051  020702' 060416  060416
2052  020706' 070416  057230  057231          .WORD   70416, 57230, 57231, 43234,   404, 70401, 76610, 76611
2053  020714' 043234  000404  070401
2054  020722' 076610  076611
2055  020726' 076614  076612  076613          .WORD   76614, 76612, 76613, 62616,120600,116174,164617, 42227
```



```
2056  020734' 062616  120600  116174
2057  020742' 164617  042227
2058  020746' 043230  076470  115204          .WORD   43230, 76470,115204,114456, 43230,   404, 76410,114457
2059  020754' 114456  043230  000404
2060  020762' 076410  114457
2061  020766' 120600  116210  120620          .WORD  120600,116210,120620,106101,104773, 70201, 40620,106502
2062  020774' 106101  104773  070201
2063  021002' 040620  106502
2064  021006' 000531  063237  000600          .WORD    531, 63237,   600, 63236, 62223,   600, 60710,114543
2065  021014' 063236  062223  000600
2066  021022' 060710  114543
2067  021026' 000403  070401  040620          .WORD    403, 70401, 40620,117635,104475,   400, 70401, 43230
2068  021034' 117635  104475  000400
2069  021042' 070401  043230
2070  021046' 000440  062710  104773          .WORD    440, 62710,104773, 60602,117246,104556, 23024,   407
2071  021054' 060602  117246  104556
2072  021062' 023024  000407
2073  021066' 073264  053221  105554          .WORD   73264, 53221,105554, 64201,   510, 43235, 62715,104505
2074  021074' 064201  000510  043235
2075  021102' 062715  104505
2076  021106' 000600  062231  023260          .WORD    600, 62231, 23260, 20640,117263, 62211, 20640, 61620
2077  021114' 020640  117263  062211
2078  021122' 020640  061620
2079  021126' 117266  020620  060360          .WORD  117266, 20620, 60360,115675,104603, 63160,115263, 20640
2080  021134' 115675  104603  063160
2081  021142' 115263  020640
2082  021146' 116277  062210  020640          .WORD  116277, 62210, 20640,116702, 20600, 60360,115710,104604
2083  021154' 116702  020600  060360
2084  021162' 115710  104604
2085  021166' 063160  115277  000405          .WORD   63160,115277,   405, 61226, 40620, 61227,100453, 23220
2086  021174' 061226  040620  061227
2087  021202' 100453  023220
2088  021206' 010052  062600  000405          .WORD   10052, 62600,   405, 63320,   760, 63260,   400, 60360
2089  021214' 063320  000760  063260
2090  021222' 000400  060360
2091  021226' 101433  043220  000405          .WORD  101433, 43220,   405,100432,     0,     0,     0,     0
2092  021234' 100432  000000  000000
2093  021242' 000000  000000
2094  021246' 000000  000000  000000          .WORD       0,     0,     0,     0,     0,     0,     0,     0
2095  021254' 000000  000000  000000
2096  021262' 000000  000000
2097  021266' 000000  000000  000000          .WORD       0,     0,     0,     0,     0,     0,     0,     0
2098  021274' 000000  000000  000000
2099  021302' 000000  000000
2100  021306' 000000  000000  000000          .WORD       0,     0,     0,     0,     0,     0,     0,     0
2101  021314' 000000  000000  000000
2102  021322' 000000  000000
2103  021326' 000000  000000  000000          .WORD       0,     0,     0,     0,     0,     0,     0,     0
2104  021334' 000000  000000  000000
2105  021342' 000000  000000
2106
2107  021346' 177777                          .WORD   -1                      ;TERMINATOR FLAG
2108  021350' 177777                          -1
```

```
                                                        .SBTTL  LISTING OF DEDICATED MICROCODE
2109
2110  021352'                                   DMAST:
2111  021352' 000400  071220  065222                    .WORD     400, 71220, 65222, 63224, 63226,   777, 63223, 63230
2112  021360' 063224  063226  000777
2113  021366' 063223  063230
2114  021372' 063231  000407  063220                    .WORD    63231,   407, 63220, 16777, 63160,101013,   600, 63222
2115  021400' 016777  063160  101013
2116  021406' 000600  063222
2117  021412' 020640  061620  103024                    .WORD    20640, 61620,103024,100431, 63170,101020, 63171,101020
2118  021420' 100431  063170  101020
2119  021426' 063171  101020
2120  021432' 104614  100440  104615                    .WORD   104614,100440,104615,   452, 23240, 60360,115732,104616
2121  021440' 000452  023240  060360
2122  021446' 115732  104616
2123  021452' 023220  010052  062600                    .WORD    23220, 10052, 62600,   405, 63320,   760, 63260,   400
2124  021460' 000405  063320  000760
2125  021466' 063260  000400
2126  021472' 060360  101433  043220                    .WORD    60360,101433, 43220,   405,100432, 33364, 24360, 53221
2127  021500' 000405  100432  033364
2128  021506' 024360  053221
2129  021512' 101465  064201  043637                    .WORD   101465, 64201, 43637,106000,113520, 23250, 63530,107620
2130  021520' 106000  113520  023250
2131  021526' 063530  107620
2132  021532' 103072  104621  060530                    .WORD   103072,104621, 60530,107217, 60603,101477,104573,120400
2133  021540' 107217  060603  101477
2134  021546' 104573  120400
2135  021552' 102173  123451  107776                    .WORD   102173,123451,107776,120620,107776, 60522,103560,123013
2136  021560' 120620  107776  060522
2137  021566' 103560  123013
2138  021572' 060602  103516  102153                    .WORD    60602,103516,102153, 60613,103543,104776,   466, 62224
2139  021600' 060613  103543  104776
2140  021606' 000466  062224
2141  021612' 000761  062225  000417                    .WORD      761, 62225,   417, 61230,120600,102124,123014,   600
2142  021620' 061230  120600  102124
2143  021626' 123014  000600
2144  021632' 060714  061220  000414                    .WORD    60714, 61220,   414, 63220,   417, 61230, 61210,   577
2145  021640' 063220  000417  061230
2146  021646' 061210  000577
2147  021652' 063262  120620  106210                    .WORD    63262,120620,106210, 60533,103546,100555,   700, 61231
2148  021660' 060533  103546  100555
2149  021666' 000700  061231
2150  021672' 000401  063302  104776                    .WORD      401, 63302,104776, 60533,103550,   576, 63262,104776
2151  021700' 060533  103550  000576
2152  021706' 063262  104776
2153  021712' 060611  103563  100570                    .WORD    60611,103563,100570, 60531,103566,100507,   600, 61231
2154  021720' 060531  103566  100507
2155  021726' 000600  061231
2156  021732' 000677  063262  100501                    .WORD      677, 63262,100501,122104,122125,   414,123000, 63260
2157  021740' 122104  122125  000414
2158  021746' 123000  063260
2159  021752' 000403  060700  061230                    .WORD      403, 60700, 61230,120600,102203,   576,123011, 60671
2160  021760' 120600  102203  000576
2161  021766' 123011  060671
2162  021772' 061220  000403  060700                    .WORD    61220,   403, 60700, 61230, 61210, 60611,103620,104553
2163  022000' 061230  061210  060611
2164  022006' 103620  104553
```



```
2165  022012' 020500  106161  120620                    .WORD    20500,106161,120620,106155, 27364, 23000,   407, 63260
2166  022020' 106155  027364  023000
2167  022026' 000407  063260
2168  022032' 000632  160400  104741                    .WORD      632,160400,104741,100726,100740,114715,104556,104556
2169  022040' 100726  100740  114715
2170  022046' 104556  104556
2171  022052' 104556  104556  063237                    .WORD   104556,104556, 63237, 60534,103317,120600,102245,120620
2172  022060' 060534  103317  120600
2173  022066' 102245  120620
2174  022072' 106064  010055  042224                    .WORD   106064, 10055, 42224, 43230, 63070, 76470,101303, 56225
2175  022100' 043230  063070  076470
2176  022106' 101303  056225
2177  022112' 041230  020640  107461                    .WORD    41230, 20640,107461,120600,102263,120620,102315, 22010
2178  022120' 120600  102263  120620
2179  022126' 102315  022010
2180  022132' 020420  103673  170617                    .WORD    20420,103673,170617, 10052, 57230, 57231, 57232, 76610
2181  022140' 010052  057230  057231
2182  022146' 057232  076610
2183  022152' 076611  062472  170617                    .WORD    76611, 62472,170617, 42225, 43230, 76470,101310,100660
2184  022160' 042225  043230  076470
2185  022166' 101310  100660
2186  022172' 041230  043230  000404                    .WORD    41230, 43230,   404, 62410,100661, 22270,104466,   442
2187  022200' 062410  100661  022270
2188  022206' 104466  000442
2189  022212' 060373  101723  100645                    .WORD    60373,101723,100645, 63467,102245,170617, 60602,103331
2190  022220' 063467  102245  170617
2191  022226' 060602  103331
2192  022232' 104564  010012  020400                    .WORD   104564, 10012, 20400,103335, 10010, 40620,106157,100767
2193  022240' 103335  010010  040620
2194  022246' 106157  100767
2195  022252' 060602  103343  104564                    .WORD    60602,103343,104564, 23010, 60530,107166, 10014,   402
2196  022260' 023010  060530  107166
2197  022266' 010014  000402
2198  022272' 060370  101761  000442                    .WORD    60370,101761,   442, 60370,101756,104556, 54620,106157
2199  022300' 060370  101756  104556
2200  022306' 054620  106157
2201  022312' 074620  040620  106157                    .WORD    74620, 40620,106157, 20420,102366,100767,106556, 30360
2202  022320' 020420  102366  100767
2203  022326' 106556  030360
2204  022332' 000402  063221  040620                    .WORD      402, 63221, 40620,101775,104570, 72601, 64201,104754
2205  022340' 101775  104570  072601
2206  022346' 064201  104754
2207  022352' 120440  103465  134620                    .WORD   120440,103465,134620,103465, 43626,107431, 60566,105440
2208  022360' 103465  043626  107431
2209  022366' 060566  105440
2210  022372' 000401  060366  105421                    .WORD      401, 60366,105421, 10060, 43234, 62574,105021,  3006
2211  022400' 010060  043234  062574
2212  022406' 105021  003006
2213  022412' 104470  070201  043234                    .WORD   104470, 70201, 43234,   576, 76674, 22760,   430,104654
2214  022420' 000576  076674  022760
2215  022426' 000430  104654
2216  022432' 104775  070204  004000                    .WORD   104775, 70204,  4000,  2777, 64201, 63537,107440,104426
2217  022440' 002777  064201  063537
2218  022446' 107440  104426
2219  022452' 000600  063302  060537                    .WORD      600, 63302, 60537,107371,104421, 63066, 63066, 63066
2220  022460' 107371  104421  063066
```

```
2221  022466' 063066  063066
2222  022472' 000724  063006  070201          .WORD     724, 63006, 70201, 64201, 16501, 62606,104771, 63066
2223  022500' 064201  016501  062606
2224  022506' 104771  063066
2225  022512' 063066  063066  063066          .WORD   63066, 63066, 63066, 63066, 63066, 63066, 63066, 63066
2226  022520' 063066  063066  063066
2227  022526' 063066  063066
2228  022532' 000640  063306  070201          .WORD     640, 63306, 70201, 54620,107371,106077,104501, 40620
2229  022540' 054620  107371  106077
2230  022546' 104501  040620
2231  022552' 107771  062606  020640          .WORD  107771, 62606, 20640,107102,   420, 60704, 62231, 70201
2232  022560' 107102  000420  060704
2233  022566' 062231  070201
2234  022572' 043230  022270  020620          .WORD   43230, 22270, 20620, 23277, 20640,106520, 20600,104513
2235  022600' 023277  020640  106520
2236  022606' 020600  104513
2237  022612' 063177  105114  020640          .WORD   63177,105114, 20640,107122, 22271, 20640, 61620,107125
2238  022620' 107122  022271  020640
2239  022626' 061620  107125
2240  022632' 020620  000775  063262          .WORD   20620,   775, 63262,   401, 62710,104771, 63066, 63066
2241  022640' 000401  062710  104771
2242  022646' 063066  063066
2243  022652' 070201  043631  106151          .WORD   70201, 43631,106151,107375,   401, 76711, 62606, 23366
2244  022660' 107375  000401  076711
2245  022666' 062606  023366
2246  022672' 100455  023366  104467          .WORD  100455, 23366,104467, 63066,104570, 63066, 63066, 63066
2247  022700' 063066  104570  063066
2248  022706' 063066  063066
2249  022712' 063066  063066  063066          .WORD   63066, 63066, 63066, 63066, 63066, 63066, 63066, 63126
2250  022720' 063066  063066  063066
2251  022726' 063066  063126
2252  022732' 000700  060706  063223          .WORD     700, 60706, 63223, 63226,120440,107771,120620,107771
2253  022740' 063226  120440  107771
2254  022746' 120620  107771
2255  022752' 023263  000603  104654          .WORD   23263,   603,104654, 60606,106371,   600, 63302,104771
2256  022760' 060606  106371  000600
2257  022766' 063302  104771
2258  022772' 063066  063066  063066          .WORD   63066, 63066, 63066, 63066, 63066, 63066, 63066, 63066
2259  023000' 063066  063066  063066
2260  023006' 063066  063066
2261  023012' 063066  061226  061207          .WORD   63066, 61226, 61207,   740, 63306,   400, 63236,120620
2262  023020' 000740  063306  000400
2263  023026' 063236  120620
2264  023032' 107627  123451  107641          .WORD  107627,123451,107641, 60616,105627,   637,104654, 63176
2265  023040' 060616  105627  000637
2266  023046' 104654  063176
2267  023052' 104627  060522  107644          .WORD  104627, 60522,107644,104631, 60531,107647,104631,   600
2268  023060' 104631  060531  107647
2269  023066' 104631  000600
2270  023072' 061231  000677  063262          .WORD   61231,   677, 63262,104627, 63237, 60606, 61223,   466
2271  023100' 104627  063237  060606
2272  023106' 061223  000466
2273  023112' 062224  000761  062225          .WORD   62224,   761, 62225,   417, 61230,120600,106265,123052
2274  023120' 000417  061230  120600
2275  023126' 106265  123052
2276  023132' 000500  063272  000600          .WORD     500, 63272,   600, 60712, 61222,   414, 63220,   417
```



```
2277  023140' 060712  061222  000414
2278  023146' 063220  000417
2279  023152' 061230  061210  060532          .WORD   61230, 61210, 60532,107707,   500, 63302,164617,   600
2280  023160' 107707  000500  063302
2281  023166' 164617  000600
2282  023172' 061231  000677  063262          .WORD   61231,   677, 63262,164617, 63236, 10006, 42224, 43230
2283  023200' 164617  063236  010006
2284  023206' 042224  043230
2285  023212' 063070  076470  105330          .WORD   63070, 76470,105330, 56225, 57220, 63060, 61210,170616
2286  023220' 056225  057220  063060
2287  023226' 061210  170616
2288  023232' 042225  043230  076470          .WORD   42225, 43230, 76470,105335,104724, 43220,   404, 76400
2289  023240' 105335  104724  043220
2290  023246' 000404  076400
2291  023252' 104725  060602  107163          .WORD  104725, 60602,107163, 23030,   405, 60370,105750,104560
2292  023260' 023030  000405  060370
2293  023266' 105750  104560
2294  023272' 030360  002402  053221          .WORD   30360,  2402, 53221, 64201, 16600, 36760, 36420, 36400
2295  023300' 064201  016600  036760
2296  023306' 036420  036400
2297  023312' 036500  036520  123210          .WORD   36500, 36520,123210,   414, 76670,  2530, 10060, 16401
2298  023320' 000414  076670  002530
2299  023326' 010060  016401
2300  023332' 104776  123230  000556          .WORD  104776,123230,   556, 60670, 61231, 23366, 60602,112402
2301  023340' 060670  061231  023366
2302  023346' 060602  112402
2303  023352' 020640  114527  004002          .WORD   20640,114527,  4002, 10004, 57234, 57233, 57235, 56227
2304  023360' 010004  057234  057233
2305  023366' 057235  056227
2306  023372' 123230  000500  063270          .WORD  123230,   500, 63270, 40710, 61231,   421, 63220, 10006
2307  023400' 040710  061231  000421
2308  023406' 063220  010006
2309  023412' 060614  112026  112426          .WORD   60614,112026,112426,   425,100642, 10006, 23650,112503
2310  023420' 000425  100642  010006
2311  023426' 023650  112503
2312  023432' 120600  112030  022202          .WORD  120600,112030, 22202, 63075, 42226, 20640,112515, 22203
2313  023440' 063075  042226  020640
2314  023446' 112515  022203
2315  023452' 061210  063075  111057          .WORD   61210, 63075,111057, 62615, 63165,111020, 10012, 43230
2316  023460' 062615  063165  111020
2317  023466' 010012  043230
2318  023472' 120600  112050  120620          .WORD  120600,112050,120620,106064, 62570,101055,114604, 76615
2319  023500' 106064  062570  101055
2320  023506' 114604  076615
2321  023512' 120600  112060  043230          .WORD  120600,112060, 43230, 62470,111070, 42227, 10006,110444
2322  023520' 062470  111070  042227
2323  023526' 010006  110444
2324  023532' 056227  043230  000404          .WORD   56227, 43230,   404, 63010, 62610,123231,   501, 60671
2325  023540' 063010  062610  123231
2326  023546' 000501  060671
2327  023552' 060710  061231  110466          .WORD   60710, 61231,110466, 61620,113106,114531,106617,120620
2328  023560' 061620  113106  114531
2329  023566' 106617  120620
2330  023572' 106064  060530  107620          .WORD  106064, 60530,107620,113026,104621, 61620,113035,114531
2331  023600' 113026  104621  061620
2332  023606' 113035  114531
```

```
2333  023612' 000522  104714  043230          .WORD     522,104714, 43230,120600,112123,120620,106050,170610
2334  023620' 120600  112123  120620
2335  023626' 106050  170610
2336  023632' 002550  100465  064201          .WORD    2550,100465, 64201, 60611,112145,112545, 23374,   424
2337  023640' 060611  112145  112545
2338  023646' 023374  000424
2339  023652' 063236  000543  063237          .WORD    63236,   543, 63237, 63176,101251,   402, 63302,114553
2340  023660' 063176  101251  000402
2341  023666' 063302  114553
2342  023672' 060602  113155  016564          .WORD    60602,113155, 16564,  2406,110565, 16647,   415, 63236
2343  023700' 002406  110565  016647
2344  023706' 000415  063236
2345  023712' 062476  070220  036400          .WORD    62476, 70220, 36400,110672, 74620, 43232, 63072, 72472
2346  023720' 110672  074620  043232
2347  023726' 063072  072472
2348  023732' 004000  020640  112171          .WORD     4000, 20640,112171,   771, 60412, 61620, 62230, 20640
2349  023740' 000771  060412  061620
2350  023746' 062230  020640
2351  023752' 112177  022010  020640          .WORD   112177, 22010, 20640,112202, 22030, 20640,112605, 20600
2352  023760' 112202  022030  020640
2353  023766' 112605  020600
2354  023772' 111624  036400  022420          .WORD   111624, 36400, 22420,   431, 60372,111617,100465, 10011
2355  024000' 000431  060372  111617
2356  024006' 100465  010011
2357  024012' 004002  016627  002776          .WORD     4002, 16627,  2776,100465, 21006, 21027,104446, 74620
2358  024020' 100465  021006  021027
2359  024026' 104446  074620
2360  024032' 043232  063072  072472          .WORD    43232, 63072, 72472,  4001, 36400, 22420,   741, 60372
2361  024040' 004001  036400  022420
2362  024046' 000741  060372
2363  024052' 111642  100465  004620          .WORD   111642,100465,  4620, 63302, 30360, 22660,100465, 10005
2364  024060' 063302  030360  022660
2365  024066' 100465  010005
2366  024072' 040620  112274  010015          .WORD    40620,112274, 10015, 43220, 76560,111260, 43220, 62560
2367  024100' 043220  076560  111260
2368  024106' 043220  062560
2369  024112' 043220  063160  105445          .WORD    43220, 63160,105445, 10011, 16725, 43232, 72472, 20400
2370  024120' 010011  016725  043232
2371  024126' 072472  020400
2372  024132' 106047  076620  022420          .WORD   106047, 76620, 22420,100465,113277,   420,110700,   421
2373  024140' 100465  113277  000420
2374  024146' 110700  000421
2375  024152' 063220  020640  107461          .WORD    63220, 20640,107461, 62210, 10015,   600, 63302, 22010
2376  024160' 062210  010015  000600
2377  024166' 063302  022010
2378  024172' 020640  112310  022030          .WORD    20640,112310, 22030, 20640,112313, 56230, 20640,112316
2379  024200' 020640  112313  056230
2380  024206' 020640  112316
2381  024212' 042230  070204  004000          .WORD    42230, 70204,  4000, 22660,100465, 16664, 43232, 63072
2382  024220' 022660  100465  016664
2383  024226' 043232  063072
2384  024232' 072472  023013  063133          .WORD    72472, 23013, 63133, 63133,   414, 62673,   453, 60372
2385  024240' 063133  000414  062673
2386  024246' 000453  060372
2387  024252' 111745  000420  060372          .WORD   111745,   420, 60372,111753,100465, 10011, 16767,  2432
2388  024260' 111753  100465  010011
```



```
2389  024266' 016767  002432
2390  024272' 000744  063237  100673          .WORD     744, 63237,100673, 10013, 36760, 23030,   407, 62670
2391  024300' 010013  036760  023030
2392  024306' 000407  062670
2393  024312' 070201  043231  060610          .WORD    70201, 43231, 60610,103465,   402, 62711,100465, 74620
2394  024320' 103465  000402  062711
2395  024326' 100465  074620
2396  024332' 043232  063072  072472          .WORD    43232, 63072, 72472,  4000, 36400, 22420,   447, 60372
2397  024340' 004000  036400  022420
2398  024346' 000447  060372
2399  024352' 115402  100465  000500          .WORD   115402,100465,   500, 62230, 10004,  4002, 43231, 56230
2400  024360' 062230  010004  004002
2401  024366' 043231  056230
2402  024372' 020640  116010  043630          .WORD    20640,116010, 43630, 62230, 23373,   402, 60370,115426
2403  024400' 062230  023373  000402
2404  024406' 060370  115426
2405  024412' 000630  063233  060531          .WORD      630, 63233, 60531,117026,   514, 63233, 60611, 61620
2406  024420' 117026  000514  063233
2407  024426' 060611  061620
2408  024432' 116433  000446  063013          .WORD   116433,   446, 63013, 60611,116436,114452, 10037,  4000
2409  024440' 060611  116436  114452
2410  024446' 010037  004000
2411  024452' 043232  004001  000423          .WORD    43232,  4001,   423, 73013, 74531,117125, 62612,   412
2412  024460' 073013  074531  117125
2413  024466' 062612  000412
2414  024472' 070013  062612  070213          .WORD    70013, 62612, 70213,  4001, 43235, 57237,   437, 63677
2415  024500' 004001  043235  057237
2416  024506' 000437  063677
2417  024512' 062230  020640  116061          .WORD    62230, 20640,116061, 56230, 63177,115061, 20640,107461
2418  024520' 056230  063177  115061
2419  024526' 020640  107461
2420  024532' 000402  063237  010034          .WORD     402, 63237, 10034,  4000, 57233, 57220, 60573,115101
2421  024540' 004000  057233  057220
2422  024546' 060573  115101
2423  024552' 063160  062230  020640          .WORD    63160, 62230, 20640,116102, 62210, 63577,115513,116111
2424  024560' 116102  062210  063577
2425  024566' 115513  116111
2426  024572' 114474  000401  114514          .WORD   114474,   401,114514,   405, 63234, 20640,116115, 56230
2427  024600' 000405  063234  020640
2428  024606' 116115  056230
2429  024612' 063174  115115  060477          .WORD    63174,115115, 60477,116474,110532,   407,114450, 61620
2430  024620' 116474  110532  000407
2431  024626' 114450  061620
2432  024632' 103055  120620  106064          .WORD   103055,120620,106064, 23620,107613, 33364,  4000, 53221
2433  024640' 023620  107613  033364
2434  024646' 004000  053221
2435  024652' 105776  064201  060600          .WORD   105776, 64201, 60600, 61620, 61620, 61620, 61620, 63226
2436  024660' 061620  061620  061620
2437  024666' 061620  063226
2438  024672' 000407  063266  104457          .WORD     407, 63266,104457, 60615, 61620, 61620, 61620, 61620
2439  024700' 060615  061620  061620
2440  024706' 061620  061620
2441  024712' 061620  063230  000407          .WORD    61620, 63230,   407, 63270, 63170,115572, 23651,116574
2442  024720' 063270  063170  115572
2443  024726' 023651  116574
2444  024732' 020600  114564  000540          .WORD    20600,114564,   540,114605, 61620,117177,114531,106617
```

```
2445  024740' 114605  061620  117177
2446  024746' 114531  106617
2447  024752' 060531  107620  117166          .WORD   60531,107620,117166,104621,   537, 63237, 10013, 57230
2448  024760' 104621  000537  063237
2449  024766' 010013  057230
2450  024772' 057231  057225  057233          .WORD   57231, 57225, 57233,120600,116213,123634,106064, 42224
2451  025000' 120600  116213  123634
2452  025006' 106064  042224
2453  025012' 056226  042225  056227          .WORD   56226, 42225, 56227, 43220, 60460, 61230,   500, 60674
2454  025020' 043220  060460  061230
2455  025026' 000500  060674
2456  025032' 060700  061231  120600          .WORD   60700, 61231,120600,116232,120620,106063,   600, 23036
2457  025040' 116232  120620  106063
2458  025046' 000600  023036
2459  025052' 060716  062223  060536          .WORD   60716, 62223, 60536,117645,115302, 10060,  2401, 60610
2460  025060' 117645  115302  010060
2461  025066' 002401  060610
2462  025072' 062222  000421  061230          .WORD   62222,   421, 61230, 60536,107460,   407, 63676, 60751
2463  025100' 060536  107460  000407
2464  025106' 063676  060751
2465  025112' 107462  000422  060416          .WORD  107462,   422, 60416, 60416, 70416, 57230, 57231, 43234
2466  025120' 060416  070416  057230
2467  025126' 057231  043234
2468  025132' 010006  076610  076611          .WORD   10006, 76610, 76611, 76614, 76605, 76613, 62616,120600
2469  025140' 076614  076605  076613
2470  025146' 062616  120600
2471  025152' 116277  164617  070201          .WORD  116277,164617, 70201, 40620,106465,   536, 63237,   600
2472  025160' 040620  106465  000536
2473  025166' 063237  000600
2474  025172' 063236  062223  000600          .WORD   63236, 62223,   600, 60710,114650, 60602,117320,104564
2475  025200' 060710  114650  060602
2476  025206' 117320  104564
2477  025212' 023024  000407  073264          .WORD   23024,   407, 73264, 53221,105562, 64201,   510, 43235
2478  025220' 053221  105562  064201
2479  025226' 000510  043235
2480  025232' 062715  104470  000600          .WORD   62715,104470,   600, 62231, 23260, 20640,117335, 62211
2481  025240' 062231  023260  020640
2482  025246' 117335  062211
2483  025252' 020640  061620  117340          .WORD   20640, 61620,117340, 20620, 60360,115747,104611, 63160
2484  025260' 020620  060360  115747
2485  025266' 104611  063160
2486  025272' 115335  020640  116351          .WORD  115335, 20640,116351, 62210, 20640,116754, 20600, 60360
2487  025300' 062210  020640  116754
2488  025306' 020600  060360
2489  025312' 115762  104612  063160          .WORD  115762,104612, 63160,115351,   405, 61226, 40620, 61227
2490  025320' 115351  000405  061226
2491  025326' 040620  061227
2492  025332' 100465  000000  000000          .WORD  100465,     0,     0,     0,     0,     0,     0,     0
2493  025340' 000000  000000  000000
2494  025346' 000000  000000
2495
2496  025352' 177777                          .WORD   -1                      ;TERMINATOR FLAG
2497  025354' 177777                          -1
2498
2499  025356' 000240                  ALDONE: NOP             ;LAST LOCATION USED BY THIS MODULE
2500
```



```
2501
2502          000001                            .END
```

```
AA11K   000260R          282#    547
ACSR    000102R          233#    433*    614     1523*
ADDR    000006R          199#    363
ADDR22= 001000           246#
AD1DRL= 000002           1576#   1587    1592    1599    1604    1611    1619    1628    1636    1648    1656    1664    1672
AD1SRH= 000001           1581#   1651    1659    1667    1675
AD1SRL= 000000           1571#   1587    1592    1599    1604    1611    1619    1628    1636    1648    1656    1664    1672
AD11K   000252R          279#    529
AD11K2  000262R          283#    542
AD2DRL= 000042           1580#   1614    1622    1631    1639    1651    1659    1667    1675
AD2SRH= 000041           1582#   1651    1659    1667    1675
AD2SRL= 000040           1579#   1614    1622    1631    1639    1648    1656    1664    1672
ALDONE  025356R          2499#
AR11K   000274R          288#    500     533     551
ASR     000106R          237#    436*
ASTAT   000104R          235#    1524*
AWAS    000110R          238#    435*
BEGIN   000000R          196#    397     398     440     461     462     470     471     477     486     510     513     521
                         523     596     607     608     618     627     634     641     642     643     644     645     646
                         647     648     649     650     651     652     653     654     655     656     657     658     659
                         660     661     960     978     983     989     1000    1006    1021    1029    1044    1065    1084
                         1530    1531
BIT0    = 000001         246#    527     603     707     969     999
BIT1    = 000002         246#    494     715     756     786     791     815     841     867     893     919
BIT10   = 002000         246#    498     515     531     549     707     783
BIT11   = 004000         246#    465
BIT12   = 010000         246#    535     707
BIT13   = 020000         246#    503
BIT14   = 040000         246#    458     554     783     1527    1534
BIT15   = 100000         246#    465     564     807
BIT2    = 000004         246#    560     807
BIT3    = 000010         246#    545     735     776     783     786
BIT4    = 000020         246#    540     736     747
BIT5    = 000040         246#    570     833
BIT6    = 000100         246#    404     976     977     998     1525    1526
BIT7    = 000200         246#    468     481     575     609     715     756     786     791     815     841     859     867
                         893     919     1081
BIT8    = 000400         246#    580     711     751     885
BIT9    = 001000         246#    585     786     911
BREAKS= 104407           246#    461     462     470     471     607     608     1530    1531
BR1     000012R          201#    973     975
BR2     000013R          202#
BTODS = 104421           246#
BUFFER  007326R          296     640     1705#
BUFF0   007326R          299     1706#
BUFF1   010330R          302     1707#
BUFF2   012334R          305     1709#
BUFF3   013336R          308     1710#
CAINC = 000000           1570#   1592    1596    1604    1608    1619    1622    1636    1639    1656    1672
CDATAS= 104412           246#
CLR     = 000000         1574#   1587    1592    1599    1604    1611    1614    1619    1622    1628    1631    1636    1639
CONFIG  000056R          221#
CSRA    000100R          231#    1522*
DATCKS= 104411           246#
DATERS= 104404           246#    440
DEVLST  006470R          493     590*    591*    594     1536#
```



```
DJOB0   002502R          707#
DJOB1   002732R          708     733     747#
DJOB2   003146R          748     774     783#
DJOB3   003314R          784     805#
DJOB4   003470R          831#
DJOB5   003644R          857#
DJOB6   004020R          883#
DJOB7   004174R          909#
DMAST   021352R          410     1719    2110#
DMDSIZ= 000022           1568#   1587    1592    1599    1604    1611    1619    1628    1636    1648    1656    1664    1672
DMDT    006516R          1538#
DONEC   007324R          943     1703#
DR11K1  000256R          281#    562
DR11K2  000264R          284#    572
DR11K3  000266R          285#    577
DR11K4  000270R          286#    582
DR11K5  000272R          287#    587
DVID1   000014R          203#
D.OCQ   006607R          1603#   1609
D.OCS   006564R          1598#   1602
D.QEQ   006541R          1591#   1597
D.QES   006516R          1586#   1590
D.TCQ   006723R          1635#   1643
D.TCQP  007037R          1671#
D.TCS   006700R          1627#   1634
D.TCSP  007014R          1663#   1670
D.TEQ   006655R          1618#   1626
D.TEQP  006771R          1655#   1662
D.TES   006632R          1610#   1617
D.TESP  006746R          1647#   1654
EBUFF0  000322R          301#    675     725     726     727
EBUFF1  000330R          304#    676     767     768     769
EBUFF2  000336R          307#    677     826     827     828     852     853     854     878     879     880     904     905
                         906     930     931     932
EBUFF3  000344R          310#    678     801     802     803
EJOB0   000352R          313#
EJOB0U  000432R          337#    679     724
EJOB1   000360R          316#
EJOB1U  000440R          340#    680     766
EJOB2   000366R          319#
EJOB2U  000446R          343#    681     800
EJOB3   000374R          322#
EJOB3U  000454R          346#    682     825
EJOB4   000402R          325#
EJOB4U  000462R          349#    683     851
EJOB5   000410R          328#
EJOB5U  000470R          352#    684     877
EJOB6   000416R          331#
EJOB6U  000476R          355#    685     903
EJOB7   000424R          334#
EJOB7U  000504R          358#    686     929
ELOOP   004350R          742     779     941#
ENDITS= 104413           246#    1021
ENDS  = 104410           246#    398     513     523
ERRTYP  000106R          236#    396*    438*    474*    483*    508*    519*    616*    625*    632*    1027*   1042*   1063*
EXITS = 104400           246#    978     1000    1084
```

```
GBUFF0  000320R   300#   716    717    718
GBUFF1  000326R   303#   757    758    759
GBUFF2  000334R   306#   816    817    818    842    843    844    868    869    870    894    895    896
                  920    921    922
GBUFF3  000342R   309#   792    793    794
GETADR  000310R   296#   595*   596    640*   641    959*   960    988*   989
GETEAR  000314R   298#   599*   600*   602    964*   965*   967    991*   992*   994
GETPAR  000312R   297#   597    961    990
GETPA$= 104415    246#   596    641    642    643    644    645    646    647    648    649    650    651
                  652    653    654    655    656    657    658    659    660    661    960    989
GJOB0   000350R   312#   1035
GJOB0U  000430R   336#   714    1055
GJOB1   000356R   315#
GJOB1U  000436R   339#   755    1039
GJOB2   000364R   318#
GJOB2U  000444R   342#   790
GJOB3   000372R   321#
GJOB3U  000452R   345#   814
GJOB4   000400R   324#
GJOB4U  000460R   348#   840
GJOB5   000406R   327#
GJOB5U  000466R   351#   866
GJOB6   000414R   330#
GJOB6U  000474R   354#   892
GJOB7   000422R   333#
GJOB7U  000502R   357#   918    1059
GMBUF$= 104414    246#
HRDCNT  000044R   216#
HRDER$= 104405    246#   477    486    510    521    618    627    634    1029   1044   1065
HRDPAS  000050R   218#
ICON1   000036R   213#
ICOUNT  000040R   214#
IDNUM   000122R   243#
IEX     004702R   981    1001#
IMAGE = ****** G    1#
INIT    000030R   210#
INPSRV  004562R   974    980#
INTR    000120R   242#
JOB0    005240R   311    697    709    946    948    1091#
JOB0U   005332R   335    713*   1093   1134#
JOB1    005334R   314    749    947    1141#
JOB1U   005426R   338    754*   1143   1184#
JOB2    005430R   317    785    1191#
JOB2U   005522R   341    789*   1193   1234#
JOB3    005524R   320    809    1241#
JOB3U   005616R   344    813*   1243   1284#
JOB4    005620R   323    835    1291#
JOB4U   005712R   347    839*   1293   1334#
JOB5    005714R   326    861    1341#
JOB5U   006006R   350    865*   1343   1384#
JOB6    006010R   329    887    1391#
JOB6U   006102R   353    891*   1393   1434#
JOB7    006104R   332    913    1441#
JOB7U   006176R   356    917*   1443   1484#
KMAD0   000230R   259#   363*   368    413*   415*   417*   418*   424*   429*   458*   465*   1527*  1534*
KMAD1   000232R   262#   369
```



```
KMAD2   000234R   264#
KMAD3   000236R   266#
KMAD4   000240R   268#   414*   419*   422*   430*   434
KMAD5   000242R   270#
KMAD6   000244R   272#   416*   425    433    435
KMAD7   000246R   274#   373
KMCSR   000230R   249#
KWT     006366R   958    1517#
KW1PRF  006372R   696*   740*   778*   1519#
KW11K   000254R   280#   496
LOPC  = 000072    248#
LPADH   000242R   269#
LPADL   000240R   267#   598*   962*   990*   1011   1034
LPCI    000230R   258#   468    601*   602*   603*   609    966*   967*   969*   976*   993*   994*   998*
                  999*   1524   1526#
LPCO    000234R   263#   481    622    977*   1008   1081*  1522   1523   1525#
LPMR    000232R   261#
LPMS1   000244R   271#
LPMS2   000246R   273#
LPSO    000236R   265#   1012   1014
LPS11   000276R   289#   505    537    556    566
MAP22$= 104416    246#
MBUFF0  000316R   299#   642
MBUFF1  000324R   302#   643
MBUFF2  000332R   305#   644
MBUFF3  000340R   308#   645
MJOB0   000346R   311#   646
MJOB0U  000426R   335#   654
MJOB1   000354R   314#   647
MJOB1U  000434R   338#   655
MJOB2   000362R   317#   648
MJOB2U  000442R   341#   656
MJOB3   000370R   320#   649
MJOB3U  000450R   344#   657
MJOB4   000376R   323#   650
MJOB4U  000456R   347#   658
MJOB5   000404R   326#   651
MJOB5U  000464R   350#   659
MJOB6   000412R   329#   652
MJOB6U  000472R   353#   660
MJOB7   000420R   332#   653
MJOB7U  000500R   356#   661
MMAST   015346R   292    406    1717   1721#
MODNAM  000000R   197#
MODREV  000224R   246#
MODSP   000224R   211    244#
MSGN$ = 104403    246#   397
MSGS$ = 104402    246#
MSGS  = 104401    246#
NER     005022R   1013   1033#
NULL  = 000000    246#   477    486    510    521    618    627    634    1029   1044   1065
OPEN  = 000000    198    204    205    206    207    224    225    226    227    228    229    230    231
                  233    235    237    238    240    241    242    246#
OTOA$ = 104420    246#
OUPEX   005234R   1083#
OUPSRV  004704R   972    1004#
```

```
OUTCON  005226R          1019    1079    1081#
PASCNT  000034R           212#
PIRQS = 000004            246#    983    1006
PUPSP = 005726            246#
PUPSP2= 022626            246#
PRIY  = 000000            246#
PRTY0 = 000000            202     246#
PRTY1 = 000040            246#
PRTY2 = 000100            246#
PRTY3 = 000140            246#
PRTY4 = 000200            246#
PRTY5 = 000240            201     246#
PRTY6 = 000300            246#
PRTY7 = 000340            246#
PS    = 177776            246#
PSW   = 177776            246#
PUSH  = 005746            246#
PUSH2 = 024646            246#
RAND$ = 104417            246#
RANNUM  000054R           220#
RESET0  006374R           475     484     613     624    1020    1026    1041    1062    1522#
RESIRT  000506R           239     362#
RES1    000056R           222#
RES2    000060R           223#
RINPR = 000015           1577#   1587    1592    1599    1604    1611    1614    1619    1622    1628    1631    1636    1639
                         1648    1651    1659    1664    1667    1675
RUNPR = 000035           1573#   1587    1592    1599    1604    1611    1614    1619    1622    1628    1631    1636    1639
                         1648    1656    1664    1672
RUNPRL= 000235           1575#   1587    1592    1599    1604    1611    1614    1619    1622    1628    1631    1636    1639
                         1651    1659    1667    1675
RSTRT   000112R           239#
SBADR   000102R           232#    437#
SCS   = 000040           1578#   1599    1604    1628    1631    1636    1639    1664    1672
SDT   = 000100           1569#   1587    1592    1599    1604    1611    1614    1622    1631    1639    1651    1659    1667
                         1675
SEN   = 000020           1583#   1648    1656
SEX   = 000100           1572#   1587    1592    1611    1614    1619    1622    1648    1656
SUFCNT  000042R           215#
SUFEN$= 104406            246#
SUFPAS  000046R           217#
SPOINT  000032R           211#
SPSIZ = 000040              1#    244
SR1     000016R           204#    394     404     494     498     503     515     517     527     531     535     540     545
                          549     554     560     564     570     575     580     585     707     736     747     783     807
                          833     859     885     911
SR2     000020R           205#    380
SR2VAL  000226R           248#    383#    740     778
SR3     000022R           206#
SR4     000024R           207#
START   000506R           210     361#    452     479     488     620     630     636    1023    1031    1046    1067
STAT    000026R           209#
STPCOD= 000071            248#
SVR0    000062R           224#
SVR1    000064R           225#
SVR2    000066R           226#
SVR3    000070R           227#
```



```
SVR4    000072R           228#
SVR5    000074R           229#
SVR6    000076R           230#
SYSCNT  000052R           219#
TABLE   007202R          1047*   1053    1691#
TABLE0  002422R           662     675#
TODOC   007200R           949*    980     996*   1683#
TODOO   007060R           941     945     985    1682#
TKPDFD= 000022            246#
TST1    000626R           395     404#
TST2    001056R           428     457#
TST3    001222R           482     493#
TST3A   001330R           497     502     507     515#
TST3B   001370R           516     518     526#
TST3C   001750R           594#
TST4    002164R           623     640#
TST4A   002454R           664     696#
UCODE   000300R           291#    408*    412*    732     773
UCODES  000302R           292#    407#    411*    423
USWH  = 000007            246#
VBM   = 000007            248#   1075
VECTOR  000010R           200#    971
VERSN   000250R           276#    590
WARN$S  006200R           397     699    1487#
WASADR  000104R           234#    434#
WDFR    000116R           241#
WDFO    000114R           240#
WRNMSG  006204R          1487    1489#
XFLAG   000005R           198#
$BDDAT  000304R           293#    968#   1016#   1017
$GDDAT  000306R           294#    701#    728#    764#    798#    819#    845#    871#    897#    923#    949    1017
.     = 025360R          1537#   1590#   1597#   1602#   1609#   1617#   1626#   1634#   1643#   1654#   1662#   1670#   1682#
                         1702#   1706#   1707#   1708#   1709#   1710#   1711#
```

```
BAHE    1096#  1099   1102   1105   1108   1111   1114   1117   1146#  1149   1152   1155   1158   1161   1164
        1167   1196#  1199   1202   1205   1208   1211   1214   1217   1246#  1249   1252   1255   1258   1261
        1264   1267   1296#  1299   1302   1305   1308   1311   1314   1317   1346#  1349   1352   1355   1358
        1361   1364   1367   1396#  1399   1402   1405   1408   1411   1414   1417   1446#  1449   1452   1455
        1458   1461   1464   1467
BKMOD     1#
BREAK     1#   460    470    606   1529
BTOD      1#
CKDATA    1#
DATACK    1#
DATERR    1#   439
DFSEVN    1#   246
DSEVNT    1#   246
END       1#   398    513    523
ENDIT     1#  1021
ENDMOD    1#
EQUATS    1#   246
EXIT      1#   978   1000   1083
GETPA     1#   596    641    642   643    644    645    646    647    648    649    650    651    652    653
        654    655    656    657   658    659    660    661    960    989
GTPAE     1#   595    959
GWBUFF    1#
HRDER     1#   476    485    509   520    617    626    633   1028   1043   1064
IOMOD     1#   190
IOMODP    1#
IOMODR    1#
IOMODX    1#
LOERRO  360#   475    484    613   624   1025   1041   1062
MAP22     1#
MODULE    1#   191
MOVEI     1#
MOVEM     1#
MRDA    1086#  1136   1186   1236   1286   1336   1386   1436
MSG       1#
MSGN      1#   397
MSGS      1#
NBKMOD    1#
OTOA      1#
PIRQ      1#   982   1004
RAND      1#
SBKMOD    1#
SDRM    744#   805    831    857    883    909
SOFER     1#
TOUT    278#
TRYFX1  941#   958
SCAL.     1#
SDMAST    1#  2110
SDMDI     1#  1536
SMMAST    1#  1721
.KMADR    1#   250
.KSIS     1#   364
.LOADL    1#
.LPAIN    1#
.PUTCS    1#
.RESET    1#
.UTK      1#
```



```
.SINLP    1#
.SMMAC    1#
.SOUTL    1#
.STLNW    1#
.STOUT  278#


. ABS.  000000    000
        025360    001


 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 ,XLPHC0.SEQ/SOL/CRF/DOC=DDXCOM,XLPHC0
 RUN-TIME:  5  8 .7 SECONDS
 RUN-TIME RATIO: 42/13=3.0
 CORE USED:  21K  (41 PAGES)

 DOCUMENT PAGES:  58
```

```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E971D-MC

PRODUCT NAME: CXRMAD0 RH11,70/RM03,02 S P M

PRODUCT DATE: FEB 1979

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RMA IS AN IOMOD THAT EXERCISES RM03/RM02 DISK DRIVES ON AN RH11/RH70 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RM03/RM02 WITH AN RH11/RH70 CONTROLLER

STORAGE:: RMA REQUIRES:
1. DECIMAL WORDS: 1825
2. OCTAL WORDS: 03441
3. OCTAL BYTES: 7102

3. PASS DEFINITION

ONE PASS OF THE RMA MODULE CONSISTS OF 1300 CYCLES OF THE BASIC TEST SEQUENCE (WRITE, WRITE-CHECK, READ, DATA-CHECK). THE TEST SEQUENCE WRITES 1024 WORDS, WRITE-CHECKS SAME, READS THE FIRST 256 WORDS, AND DATA-CHECKS SAME.

4. EXECUTION TIME

ONE PASS OF RMA RUNNING ALONE ON A PDP-11/40 TAKES APPROXIMATELY 0.9 MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 176700, VECTOR: 254, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

THIS MODULE ALSO SUPPORTS RP04/5/6 ON THE SAME MASSBUS CONTROLLER. HOWEVER,THIS MODULE IS NOT USED TO EXERCISE RP04/5/6 ALONE.

7. MODULE OPERATION

TEST SEQUENCE:

A. SETUP DEVICE REGISTER ADDRESSES AND MODULE VARIABLES
B. RESET ALL DRIVES ON-LINE AND DROP ALL THAT ARE NOT
C. GET A STARTING SECTOR ADDRESS
D. GET A DRIVE ADDRESS
E. DO A WRITE -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
F. DO A WRITE-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
G. DO A READ -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
H. DO A DATA-CHECK -- IF ERRORS, REPORT AND RETRY UP TO RETRY LIMIT
I. IF END OF PASS, REPORT AND GO TO C
J. IF END OF DRIVES, GO TO C ELSE GO TO D

8. OPERATION OPTIONS

SR1 BIT0 SET(1):
IF THE RETRY LIMIT IS EXCEEDED ON ANY FUNCTION, A HARD ERROR IS ASSUMED AND THE DRIVE IS DROPPED

SR1 BIT0 CLEAR(0):
IF THE RETRY LIMIT IS EXCEEDED, THE FUNCTION IS ABORTED AND THE TESTING CONTINUES

SR1 BIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

SR1 BIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM

```
SR1 BIT5 CLEAR (0)      ;NORMAL FOR RM03 PACKS THAT ARE FORMATED
                        ;FOR 16 BIT MODE (PDP-11)
SR1 BIT5 SET (1)        ;FOR RM03 18 BIT FORMATED PACKS

SR1 BIT15 SET (1)       ;32 REGISTERS ON RH70
SR1 BIT15 CLEAR (0)     ;22 REGISTERS ON RH70
```

9. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT

B. ERROR MESSAGES DUMP THE CONTENTS OF THE 20 RH11/RM03 REGISTERS IN THE FOLLOWING ORDER:

RMCS1 RMWC RMBA RMDA RMCS2 RMDS RMER1 RMAS
RMLA RMDB RMMR1 RMDT RMSN RMOF RMDC RMHR
RMMR2 RMER2 RMEC1 RMEC2 RMBAE RMCS3 XFER CNT

10. BAD SPOTS

A. LOCATION 256 THROUGH 452 CONTAIN ROOM FOR 32. BAD SPOTS.
EACH BAD SPOT TAKES TWO WORDS.
THE FIRST WORD SPECIFIES THE CYLINDER ADDRESS.
THE LOW BYTE OF THE SECOND WORD SPEPICIES THE SECTOR ADDRESS.
THE HIGH BYTE OF THE SECOND WORD SPECIFIES THE TRACK ADDRESS.

B. THE RMA MODULE DOES NOT ACCESS THESE SPOTS ON ANY OF THE DRIVE UNDER TEST.

C. THE BAD SPOT FILE IS RETRIEVED FROM ALL DRIVES ASSIGNED IN THE BIT MAP "DVID1".

(CYLINDER 822, TRACK 4, SECTOR 0 - MANUFACTUR BAD SPOT FILE)
(CYLINDER 822, TRACK 4, SECTOR 12 - USER BAD SPOT FILE)
LOCATION 1760 MUST BE SET TO 10 TO SEARCH USER BAD SPOT FILE

```
                        IOMODX  <RMAD  >,176700,254,5,0,0,1300.,144,BUFIN,256.,1024.
                        MODULE  150000,RMAD  ,176700,254,5,0,0,1300.,144,BUFIN,256.,1024.
                        .TITLE  RMAD DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION  6      23-MAY-78
                                .LIST   BIN
                        ;*****************************************************************************
000000'                 BEGIN:
000000' 046522  042101      040 MODNAM: .ASCII  /RMAD / ;MODULE NAME.
000005'     000         XFLAG:  .BYTE   0PEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 176700          ADDR:   176700+0                ;1ST DEVICE ADDR.
000010' 000254          VECTOR: 254+0                   ;1ST DEVICE VECTOR.
000012'     240         BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'     000         BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001          DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000          SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000          SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000          SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000          SR4:    OPEN                    ;SWITCH REGISTER 4
                        ;*****************************************************************************
000026' 150000          STAT:   150000                  ;STATUS WORD.
000030' 001662'         INIT:   START                   ;MODULE START ADDR.
000032' 000252'         SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000          PASCNT: 0                       ;PASS COUNTER.
000036' 002424          ICONT:  1300.                   ;# OF ITERATIONS PER PASS=1300.
000040' 000000          ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000          SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000          HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000          SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000          HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000          SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000          RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                 CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000          RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000          RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000          SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000          SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000          SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000          SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000          SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000          SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000          SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000          CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                 SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000          ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                 WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000          ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                 ERRTYP:                         ;TYPE OF ERROR
000106' 000000          ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000          AWAS:   OPEN                    ;ACTUAL DATA.
000112' 002534'         RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000          WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000          WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000          INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000144          IDNUM:  144                     ;MODULE IDENTIFICATION NUMBER=144
000124' 006600'         RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126' 000000          RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
```



```
000130' 000000          RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132' 000400          RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134' 000000          WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136' 000000          WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140' 002000          WBUFRQ: 1024.                   ;WRITE BUFFER SIZE REQUESTED
000142' 000000          WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144' 000000          CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146' 000000          CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150' 000000          FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                  .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                .NLIST
                                .WORD   0
                                .LIST
                                .ENDR
000252'                 MODSP:
                        ;*****************************************************************************
```

```
000252' 000254'                 FILE:   .WORD   BADSPT          ;BAD SECTOR TABLE ADDRESS
000254'                         BADSPT:                         ;BAD SPOT TABLE
                                                                ;FIRST WORD = CYLINDER ADDRESS
                                                                ;SECOND WORD = TRACK + SECTOR ADDRESS
000254' 177777                          .WORD   -1
000256' 177777                          .WORD   -1
000260' 177777                          .WORD   -1
000262' 177777                          .WORD   -1
000264' 177777                          .WORD   -1
000266' 177777                          .WORD   -1
000270' 177777                          .WORD   -1
000272' 177777                          .WORD   -1
000274' 177777                          .WORD   -1
000276' 177777                          .WORD   -1
000300' 177777                          .WORD   -1
000302' 177777                          .WORD   -1
000304' 177777                          .WORD   -1
000306' 177777                          .WORD   -1
000310' 177777                          .WORD   -1
000312' 177777                          .WORD   -1
000314' 177777                          .WORD   -1
000316' 177777                          .WORD   -1
000320' 177777                          .WORD   -1
000322' 177777                          .WORD   -1
000324' 177777                          .WORD   -1
000326' 177777                          .WORD   -1
000330' 177777                          .WORD   -1
000332' 177777                          .WORD   -1
000334' 177777                          .WORD   -1
000336' 177777                          .WORD   -1
000340' 177777                          .WORD   -1
000342' 177777                          .WORD   -1
000344' 177777                          .WORD   -1
000346' 177777                          .WORD   -1
000350' 177777                          .WORD   -1
000352' 177777                          .WORD   -1
000354' 177777                          .WORD   -1
000356' 177777                          .WORD   -1
000360' 177777                          .WORD   -1
000362' 177777                          .WORD   -1
000364' 177777                          .WORD   -1
000366' 177777                          .WORD   -1
000370' 177777                          .WORD   -1
000372' 177777                          .WORD   -1
000374' 177777                          .WORD   -1
000376' 177777                          .WORD   -1
000400' 177777                          .WORD   -1
000402' 177777                          .WORD   -1
000404' 177777                          .WORD   -1
000406' 177777                          .WORD   -1
000410' 177777                          .WORD   -1
000412' 177777                          .WORD   -1
000414' 177777                          .WORD   -1
000416' 177777                          .WORD   -1
000420' 177777                          .WORD   -1
000422' 177777                          .WORD   -1
```



```
000424' 177777                          .WORD   -1
000426' 177777                          .WORD   -1
000430' 177777                          .WORD   -1
000432' 177777                          .WORD   -1
000434' 177777                          .WORD   -1
000436' 177777                          .WORD   -1
000440' 177777                          .WORD   -1
000442' 177777                          .WORD   -1
000444' 177777                          .WORD   -1
000446' 177777                          .WORD   -1
000450' 177777                          .WORD   -1
000452' 177777                          .WORD   -1
000454' 000000                  DLTCNT: 0                       ;DATA LATE ERROR COUNTER
000456' 000000                  CLK:    0
000460' 000000                  BLK1:   0                       ;DISK BLOCK COUNTER
000462' 000000                  CYL:    0                       ;CALCULATED CYLINDER ADDRESS
000464' 000000                  CYLSAV: 0                       ;MODIFIED CYLINDER ADDRESS FOR RM03
000466' 000000                  SEC:    0
000470' 000000                  CNT:    0                       ;CYCLE COUNTER
000472' 000000                  CNT1:   0                       ;CYCLE COUNTER1 - DT03 SUPPORT
000474' 000000                  FUNC:   0
000476' 000000                  ONCE:   0
000500' 000000                  BADSEC: 0               ;HARDWARE DETECTOR BAD SECTOR
000502' 000000                  FERADR: 0

                                ;DO NOT CHANGE THE ORDER OF THE NEXT 15 LOCATIONS
                                ;NEEDED FOR MAP22 ROUTINE

000504' 000000                  PA18:   0
000506' 000000                  XMEM:   0
000510' 000000                  PA22:   0
000512' 000000                  EA22:   0
000514' 000000                  MBLKRV: 2                       ;HOLDS 1024 WD BLKS PER TRACK
000516' 000000                  MBLKTR: 0                       ;HOLDS BLKS PER CYLINDER
000520' 000000                  MODE:   0                       ;HOLDS LAST SECTOR # THAT WILL FIT PER TRACK
000522' 000000                  MLOWCY: 0                       ;BLKS THAT FIT IN FIRST 410 CYLINDERS (0-409)
000524' 000000                  MHICY:  0                       ;BLKS THAT FIT IN LAST 413 CYLINDERS (410-822)
                                ;

                                                                ;1 BLOCK =4 SECTORS = 1024 (256 X 4 ) WORD.
000526' 000010                  T16:    8.                      ;THESE 5 LOC. GET MOVED TO ABOVE 5 IF 16 BIT MODE
000530' 000050                          40.                     ;1024 WORD BLOCKS/CYLINDER
000532' 000040                          32.                     ;SECTORS/TRACK 8. X 4.
000534' 040020                          410.*40.                        ;410. X 40. (CYLINDER 0 - 409)
000536' 040157                          413.*40.-25.                    ;413. X 40. (CYLINDER 410 - 822)
                                                                        ;EXCLUDING THE LAST 25 BLOCKS


                                ;*****************
000540' 000007                  T18:    7.                      ;THESE 5 LOC GET MAPPED INTO FIRST 5 ABOVE IF 18 BIT MOD
000542' 000043                          35.                     ;1024 WORD (18 BIT) BLOCKS/CYLINDER
000544' 000034                          28.                     ;SECTORS/TRACK 7. X 4.
                                                                ;ACTUAL SECTORS ON A TRACK
                                                                ;FOR 18 BIT WORD FORMAT ARE 30.
000546' 034016                          410.*35.                        ;410. X 35. (CYLINDER 0 - 409)
000550' 034136                          413.*35.-25.                    ;413. X 35. (CYLINDER 410 - 822)
                                                                        ;EXCLUDING THE LAST 25 BLOCKS
```

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

000552' 000000           ZERO:   0
000554' 000000           DSKADR: 0
000556' 000000           DVICE:  0
000560' 000000           DRIVE:  0
000562' 000000           BLKSAV: 0
000564' 000000           TBUF:   0
000566' 000000           WCNT1:  0
000570' 000000           WCNT2:  0
000572' 000000           UNITNO: 0
000574' 000000           MIXDV:  0               ;MIXDV=-1,IF NOT A RM03 OR RM02
000576' 000000           MOD1:   0
000600' 000400           BUFIN:  .BLKW   256.
```



```
001600'                  TABLE:
001600' 000000           RMCS1:  0
001602' 000000           RMWC:   0
001604' 000000           RMBA:   0
001606' 000000           RMDA:   0
001610' 000000           RMCS2:  0
001612' 000000           RMDS:   0
001614' 000000           RMER1:  0
001616' 000000           RMAS:   0
001620' 000000           RMLA:   0
001622' 000000           RMDB:   0
001624' 000000           RMMR1:  0
001626' 000000           RMDT:   0
001630' 000000           RMSN:   0
001632' 000000           RMOF:   0
001634' 000000           RMDC:   0
001636' 000000           RMHR:   0
001640' 000000           RMMR2:  0
001642' 000000           RMER2:  0
001644' 000000           RMEC1:  0
001646' 000000           RMEC2:  0
001650' 000000           RMBAE:  0
001652' 000000           RMCS3:  0
001654' 000502'          XFERAD: FERADR
001656' 000470'          XFERCT: CNT
001660' 177777                   177777
```

```
001662' 012767  002000  176226  START:  MOV     #1024.,WDER     ;1024. WORDS FROM MEM/ITERATION
001670' 012767  000400  176216          MOV     #256.,WDIO      ;256 WORDS TO MEM/ITERATION
001676' 012767  000003  176214          MOV     #3,INTR         ;3 INTERRUPTS/ITERATION
001704' 005067  176560                  CLR     CNT             ; ZERO END OF PASS TESTER
001710' 005067  176540                  CLR     DLTCNT          ;CLEAR DATA LATE ERROR COUNTER
001714' 012767  010000  176654          MOV     #BIT12,MOD1     ;SETUP FOR 16 BIT MODE (NORMAL)
001722' 012700  000526'                 MOV     #T16,R0         ;GET TABLE OF VALUES FOR 16 BIT MODE
001726' 032767  000040  176062          BIT     #RIT5,SR1       ;16 BIT MODE?
001734' 001404                          BEQ     2$              ;YES
001736' 005067  176634                  CLR     MOD1            ;NO 18 BIT MODE, CLEAR FMT BIT
001742' 012700  000540'                 MOV     #T18,R0         ;GET TABLE OF VALUES FOR 18 BIT MODE
001746' 012701  000514'         2$:     MOV     #MBLERR,R1      ;BEGIN OF TABLE ENTRIES
001752' 012702  000005                  MOV     #5,R2           ;NO. OF ENTRIES
001756' 012021          3$:             MOV     (R0)+,(R1)+     ;STORE AN ENTRY
001760' 005302                          DEC     R2              ;COUNT IT
001762' 001375                          BNE     3$              ;DO MORE
001764' 012767  000007  176504  1$:     MOV     #7,ONCEE        ;SET ONE TIME ONLY FLAGS
001772' 105067  005102                  CLRB    FLAG            ; CLEAR FLAGS
001776' 016767  176012  176552          MOV     DVI01,DVICE     ; GET DRIVE INDICATOR
002004' 122737  000016  000041          CMPB    #16,@#41        ;IF RM IS LOAD MEDIUM THEN
002012' 001021                          BNE     13$             ;BEGIN
002014' 113700  000040                  MOVB    @#40,R0         ;  GET LOAD-DRIVE NUMBER
002020' 012701  000001                  MOV     #1,R1           ;  INITIALIZE DRIVE POINTER
002024' 105700          10$:            TSTB    R0              ;  WHILE NOT POINTING AT LOAD-DRIVE DO
002026' 001403                          BEQ     12$             ;  BEGIN
002030' 006301                          ASL     R1              ;    POINT TO NEXT DRIVE
002032' 105300                          DECR    R0              ;    COUNT SHIFTS
002034' 000773                          BR      10$             ;  END
002036' 130167  176514          12$:    BITB    R1,DVICE        ;  IF LOAD-DRIVE SELECTED THEN
002042' 001405                          BEQ     13$             ;  BEGIN
002044' 140167  176506                  BICR    R1,DVICE        ;    DROP THE DEVICE
002050' 104403  000000' 006746'         MSGN$,BEGIN,LDRIVE      ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        ;  END
002056'                         13$:    ;END
002056' 005767  176474                  TST     DVICE           ;IF NO DRIVES ARE SELECTED THEN
002062' 001002                          BNE     DAOST           ;BEGIN
002064' 000167  000752                  JMP     FINI            ;  DROP MODULE
                                        ;END
002070' 016767  176462  176462  DAOST:  MOV     DVICE,DRIVE     ; ALSO SAVE IT IN DRIVE
002076' 016706  175730                  MOV     SPOINT,R6       ;RESTORE STACK POINTER
002102' 012767  177777  176350          MOV     #-1,BLK1        ; INITIALIZE BLOCK COUNTER
002110' 012767  177777  176454          MOV     #-1,UNITNO      ; INITIALIZE DEVICE COUNTER
002116' 004767  003602                  JSR     PC,SETUP        ;SET UP CONTROLLER REGISTER
002122' 004767  003504                  JSR     PC,REZET        ;RESET CONTROLLER AND DRIVES
002126' 004767  003070                  JSR     PC,FDUNIT       ;FOUND THE FIRST DRIVE AVAILABLE
002132' 132767  000010  004740          BITB    #BIT3,FLAG      ;NO DRIVE ?
002140' 001402                          BEQ     .+6             ;DRIVE AVAILABLE
002142' 000167  000674                  JMP     FINI            ;DROP THE MODULE
002146' 004567  003330                  JSR     R5,READY        ;DEVICE READY ?
002152' 000402                          BR      CT              ;YES
002154' 004767  003120                  JSR     PC,NOTRDY       ;NO ,TRY NOT READY ROUTNIE
002160'                         CT:
002160' 005767  176372                  TST     DVICE           ;DROP THE MODULE ?
                                        ;       BNE     BADFL   ;NO
002164' 001002                          BNE     1$              ;BRANCH IF DRIVE AVAILABLE
002166' 000167  000650                  JMP     FINI            ;YES DROP MODULE
```



```
002172' 012701  000254'         1$:     MOV     #BADSPT,R1              ;TABLE ADDRESS
002176' 012702  000100                  MOV     #64.,R2         ;32 WORDS
002202' 012721  177777          2$:     MOV     #-1,(R1)+       ;RESET TO -1
002206' 005302                          DEC     R2              ;DECREMENT ONE WORD
002210' 001374                          BNE     2$              ;BRANCH IF NOT DONE
002212' 012767  001466  176242  BADFL:  MOV     #822.,CYL       ;RETRIEVE THE BAD SPOT FILE
002220' 112767  000004  176327          MOVB    #4,DSKADR+1     ;FROM THE PACK CYL-822, TRK-4
002226' 112767  000000  176320          MOVB    #0,DSKADR       ;SEC-0(SEC-12 FOR USER BAD SPOT FILE)
002234' 005767  176336                  TST     MOD1            ;IN 16 BIT MODE ?
002240' 001002                          BNE     .+6             ;BRANCH IF IT IS
002242' 005267  176306                  INC     DSKADR          ;OTHERWISE START FROM SECTOR 1
002246' 016777  176320  177334          MOV     UNITNO,@RMCS2   ;LOAD DRIVE NUMBER
002254' 012777  000000  177352          MOV     #0,@RMDC        ;SET TO CYL 0
002262' 012777  000000  177316          MOV     #0,@RMDA        ;SET TO TRK 0,SEC 0 AT START TIME
002270'                         BADFL2:
002270' 104415  000000' 000124'         GETPA$,BEGIN,,RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002274' 004767  002636                  JSR     PC,RM03CK       ;CHECK IF RM03 OR RM02
002300' 005767  176266                  TST     $IXDV           ;NOT THEM ?
002304' 100455                          BMI     4$              ;BRANCH IF NOT
002310' 016767  175616  176252          MOV     RBUFSZ,WCNT2    ;WORD CTR
002316' 005467  176246                  NEG     WCNT2           ;2'S COMPLEMENT
002322' 004567  001104                  JSR     R5,READ         ;READ ONE SECTOR AT A TIME
002326' 000167  000150                  JMP     RETRY           ;BRANCH IF ERROR
002332' 012701  000600'                 MOV     #BUFIN,R1       ;BUFFER ADDRESS
002336' 012702  000254'                 MOV     #BADSPT,R2              ;TABLE ADDRESS
002342' 012703  000200                  MOV     #128.,R3                ;MAX 16 BAD SECTORS, EACH SECTOR DEFINED BY TWO
002346' 060203                          ADD     R2,R3           ;LAST ADDRESS
002350' 012704  000400                  MOV     #256.,R4        ;END OF BAD SPOT FILE
002354' 060104                          ADD     R1,R4           ;LAST ADDRESS OF BAD SPOT FILE
002356' 062701  000010                  ADD     #10,R1          ;BAD SECTOR STARTS AT 5TH WORD
002362' 022711  177777          1$:     CMP     #-1,(R1)        ;TABLE IS EMPTY ?
002366' 001425                          BEQ     4$              ;BRANCH IF IT IS
002370' 022712  177777                  CMP     #-1,(R2)        ;ENTRY FOR BAD SPOT TABLE ?
002374' 001006                          BNE     2$              ;BRANCH IF NOT
002376' 011112                          MOV     (R1),(R2)       ;LOAD CYLINDER ADDRESS
002400' 016162  000002  000002          MOV     2(R1),2(R2)     ;LOAD THE TRACK AND SECTOR ADDRESSES
002406' 062701  000004                  ADD     #4,R1           ;ADJUST POINTER
002412' 062702  000004          2$:     ADD     #4,R2           ;ADJUST POINTER
002416' 020104                          CMP     R1,R4           ;END OF TABLE ?
002420' 103010                          BHIS    4$              ;BRANCH IF SO
002422' 020203                          CMP     R2,R3           ;OVER 32 BAD SECTORS RECORDED ?
002424' 103001                          BHIS    3$              ;BRANCH IF SO
002426' 000755                          BR      1$              ;LOOP BACK
002430' 104403  007044' 000000' 3$:     MSGN$,MES14,BEGIN       ;MESSAGE: OVER 32 BAD SECTORS
002436' 000167  000400                  JMP     FINI            ;DROP THE MODULES
002442' 004767  003554          4$:     JSR     PC,FDUNIT       ;FOUND NEXT DRIVE
002446' 132767  000010  004424          BITB    #BIT3,FLAG      ;DRVIE AVAILABLE ?
002454' 001006                          BNE     5$              ;BRANCH IF NOT
002456' 004567  003020                  JSR     R5,READY        ;LET DRIVE READY
002462' 000653                          BR      BADFL           ;BRANCH IF DRIVE READY
002464' 004767  002610                  JSR     PC,NOTRDY       ;TRY DRIVE NOT READY
002470' 000650                          BR      BADFL           ;BRANCH IF SUCCESSFUL
002472' 012767  177777  176072  5$:     MOV     #-1,UNITNO      ;RESET UNIT NUMBER
                                        ;ALL ON LINE DRIVE HAS BEEN RETRIVED THE BAD SPOT FILE
002500' 000425                          BR      RSTRT1          ;+ SUPPORT - DT03
```

```
002502' 105267  176046          RETRX:  INCB    DSKADR          ;ADJUST 2 SECTOR
002506' 105267  176042                  INCB    DSKADR          ;ADJUST TWO SECTOR
002512' 122767  000036  176034          CMPB    #30.,DSKADR     ;SECTORS 0,2,4,.....30, ALL TRIED
002520' 101263                          BHI     BADFL2          ;BRANCH IF NOT
002522' 104403  007050' 000000'         MSGN$,MES15,BEGIN       ;MESSAGE : RETRIEVE FAILS
002530' 000167  000306                  JMP     FINI            ;DROP THE MODULES
```



```
002534' 005767  175732          RESTRT: TST     CNT1            ;+      / SUPPORT
002540' 001005                          BNE     RSTRT1          ;+     /  FOR
002542' 005767  175722                  TST     CNT             ;+    /   DT03
002546' 001002                          BNE     RSTRT1          ;+   /    BUS
002550' 000167  177106                  JMP     START           ;+ /      SWITCH
002554'                         RSTRT1:                         ;+/       OPTION
002554' 104415  000000' 000124'         GETPAS,BEGIN, RBUFVA    ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
002562' 016767  175344  176000          MOV     RBUFSZ,WCNT2    ; SAVE READ BUFFER SIZE
002570' 005467  175774                  NEG     WCNT2           ; GET THE 2'S COMPLEMENT
002574' 016767  175660  175760  STRT:   MOV     BLK1,BLKSAV     ;INIT BLOCK COUNTER
002602'                         STRT1:
002602' 104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
002606' 016767  175330  175752          MOV     WBUFSZ,WCNT1    ; SAVE WRITE BUFFER SIZE
002614' 005467  175746          2$:     NEG     WCNT1           ; GET THE 2'S COMPLEMENT
002620' 004767  002376          NEXT:   JSR     PC,FDUNIT       ; FIND UNIT #
002624' 016777  175742  176756          MOV     UNITNO,@RMCS2   ;GET UNIT # SO BOTH PORTS
                                                                ;WILL BE LOOKING FOR SAME DRIVE
002632' 005767  175720                  TST     DVICE           ; ANY DRIVES LEFT ?
002636' 001002                          BNE     1$              ; YES
002640' 000167  000176                  JMP     FINI            ; NO DROP MODULE
002644' 132767  000010  004226  1$:     BITB    #BIT3,FLAG      ;MORE DRIVES ON SYS?
002652' 001404                          BEQ     ABW             ;YES
002654' 012767  177777  175710          MOV     #-1,UNITNO      ;RESET DRIVE NUMBER
002662' 000744                          BR      STRT            ;YES CORRECT SETUP FOR DUEL PORT RETURN
002664' 004767  002250          ABW:    JSR     PC,RM03CK       ;UPDATE DSK ADDR
002670' 004567  002606                  JSR     R5,READY        ; IS DRIVE READY ?
002674' 000402                          BR      1$              ; YES, CONTINUE
002676' 004767  002376                  JSR     PC,NOTRDY       ; NOT READY, GO WAIT UNTIL IT IS
002702' 005067  004156          1$:     CLR     TRY1            ; ZERO RETRY COUNTERS
002706' 005067  004154                  CLR     TRY3            ;
002712' 005067  004152                  CLR     TRY5
002716' 005067  004150                  CLR     TRY7
002722' 005067  004146                  CLR     TRY9
002726' 005067  004144                  CLR     TRY11
002732' 004567  000466                  JSR     R5,UPDATT               ;TO SEE IF RM03 OR RM02
                                                                        ;FOR THE SELECTED DRIVE
```

```
002736' 004567  000272          GOC:    JSR     R5,WRITE                ; WRITE SOME DATA
002742' 000167  000104                  JMP     RETRY1                  ; IF ERRORS, TRY IT AGAIN
002746' 132767  000004  004124          BITB    #BIT2,FLAG              ; DISK OVERFLOW ?
002754' 001410                          BEQ     GOA                     ; NO, CONTINUE
002756' 142767  000004  004114  DAOCL:  BICB    #BIT2,FLAG              ; YES, CLEAR OVERFLOW FLAG
002764' 012767  177777  175466          MOV     #-1,BLKI                ; RESET BLOCK #
002772' 000167  177576                  JMP     STRT                    ; CONTINUE
002776' 004567  000326          GOA:    JSR     R5,WRITCK               ; WRITE-CHECK THE DATA
003002' 000167  000114                  JMP     RETRY2                  ; IF ERRORS, TRY AGAIN
003006' 004567  000420          GOB:    JSR     R5,READ                 ; READ THE DATA WRITTEN
003012' 000167  000136                  JMP     RETRY3                  ; IF ERRORS, TRY AGAIN
003016' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA             ; REQUEST FOR MONITOR TO CHECK DATA
003024' 003026'                         .+2                             ; IF ERROR, CONTINUE
003026'                         CYCLE:
003026' 005267  175440                  INC     CNT1                    ;+BUMP INDICATOR FOR DT03
003032' 104413  000000'                 ENDIT$,BEGIN                    ;SIGNAL END OF ITERATION,
                                                                        ;MONITOR SHALL TEST END OF PASS
003036' 000167  177556                  JMP     NEXT
003042'                         FINI:
003042' 104410  000000'                 ENDS,BEGIN                      ; DROP THE MODULE
                                ;       ------------------------
003046' 000167  177546          NXT1:   JMP     NEXT                    ;GET NEXT DRIVE
003052' 005767  175422          RETRY1: TST     BADSEC                  ;BAD SECTOR DETECTED ?
003056' 001404                          BEQ     .+10.                   ;DETORE
003060' 005067  175414                  CLR     BADSEC                  ;CLEAR THE FLAG
003064' 000167  000116                  JMP     NEXTA                   ;EXIT
003070' 105267  003770                  INCB    TRY1                    ; COUNT THE RETRYS
003074' 122767  000003  003762          CMPB    #3,TRY1                 ; LIMIT EXCEEDED
003102' 001402                          BEQ     1$                      ; YES
003104' 000167  177626                  JMP     GOC                     ; NO RETRY
003110'                         1$:
003110' 104403  000000' 006766'         MSGN$,BEGIN,EXCED1              ;ASCII MESSAGE CALL WITH COMMON HEADER
003116' 000167  000064                  JMP     NEXTA                   ; GO TO NEXT DRIVE
003122' 105267  003737          RETRY2: INCB    TRY2                    ; COUNT THE RETRYS
003126' 122767  000003  003731          CMPB    #3,TRY2                 ; LIMIT EXCEEDED
003134' 001402                          BEQ     1$                      ; YES
003136' 000167  177634                  JMP     GOA                     ; NO RETRY
003142'                         1$:
003142' 104403  000000' 006774'         MSGN$,BEGIN,EXCED2              ;ASCII MESSAGE CALL WITH COMMON HEADER
003150' 000167  000032                  JMP     NEXTA                   ; GO TO NEXT DRIVE
003154' 105267  003706          RETRY3: INCB    TRY3                    ; COUNT THE RETRYS
003160' 122767  000003  003700          CMPB    #3,TRY3                 ; LIMIT EXCEEDED
003166' 001402                          BEQ     1$                      ; YES
003170' 000167  177612                  JMP     GOB                     ; NO RETRY
003174'                         1$:
003174' 104403  000000' 007002'         MSGN$,BEGIN,EXCED3              ;ASCII MESSAGE CALL WITH COMMON HEADER
003202' 000167  000000                  JMP     NEXTA                   ; GO TO NEXT DRIVE
```



```
003206' 032767  000001  174602  NEXTA:  BIT     #BIT0,SR1               ; DROP THE DRIVE
003214' 001405                          BEQ     1$                      ; NO, SKIP TO NEXT DRIVE
003216' 004767  000634                  JSR     PC,DROP                 ; YES, DROP OFFENDING DRIVE
003222' 104403  000000' 007024'         MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
003230' 000167  177364          1$:     JMP     NEXT                    ; GO ON TO NEXT DRIVE
                                ;       ------------------------


                                        .MACRO  LINEUP  EABITS  ; LINE UP EA BITS FOR RHCS1
                                ;       LINEUP  EABITS
                                        MOV     EABITS,R0               ; GET EXTENDED MEMORY BITS
                                        ASL     R0                      ; SHIFT 4 PLACES TO THE LEFT
                                        ASL     R0                      ; TO LINE UP WITH RHCS1
                                        ASL     R0                      ;
                                        ASL     R0                      ;
                                        MOV     R0,XMEM                 ; SAVE THE SHIFTED BITS
                                        .ENDM   LINEUP
```

```
                                        ;               ----- RM DISK DRIVERS -----
                                        ;
003234' 012767  000161  175232  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
003240' 016777  175320  176332          MOV     WCNT,@RMWC      ; LOAD WORD COUNT
003246' 016777  174660  176326          MOV     RBUFF1,@RMBA    ; LOAD BUFFER ADDRESS
003256' 016777  175272  176322          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003264' 004767  001650                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003270' 016777  175166  176336          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003276' 016767  175162  175156          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  RBUFEA
003304' 016700  174626                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
003310' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003312' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003314' 006300                          ASL     R0              ;
003316' 006300                          ASL     R0              ;
003320' 010067  175162                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003324' 000167  000300                  JMP     GOGO            ; CONTINUE
003330' 012767  000151  175136  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
003336' 016777  175224  176236          MOV     WCNT1,@RMWC     ; LOAD WORD COUNT
003344' 016777  174564  176232          MOV     RBUFF1,@RMBA    ; LOAD BUFFER ADDRESS
003352' 016777  175176  176226          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003360' 004767  001554                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003364' 016777  175072  176242          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003372' 016767  175066  175062          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  RBUFEA
003400' 016700  174532                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
003404' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003406' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003410' 006300                          ASL     R0              ;
003412' 006300                          ASL     R0              ;
003414' 010067  175066                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003420' 000167  000204                  JMP     GOGO            ; CONTINUE
003424' 004767  001010  UPDATT: JSR     PC,BLOCK
003430' 000205                          RTS     R5
                                        ;
003432' 012767  000171  175034  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
003440' 016777  175124  176134          MOV     WCNT,@RMWC      ; LOAD WORD COUNT
003446' 016777  174464  176130          MOV     RBUFF1,@RMBA    ; LOAD BUFFER ADDRESS
003454' 016777  175074  176124          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003462' 004767  001452                  JSR     PC,RM03CK       ; CHECK FOR RM03 DRIVE
003466' 016777  174770  176140          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
003474' 016767  174764  174760          MOV     CYLSAV,CYL      ; RESTORE CALCULATED CYLINDER ADDRESS
                                        ;       LINEUP  RBUFEA
003502' 016700  174422                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
003506' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003510' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003512' 006300                          ASL     R0              ;
003514' 006300                          ASL     R0              ;
003516' 010067  174764                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003522' 000442                          BR      GOGO            ; CONTINUE
```



```
003524' 016777  175042  176056  CLEAR:  MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
003532' 012777  000011  176040          MOV     #11,@RMCS1      ; ISSUE A DRIVE CLEAR
003540' 000240                          NOP                     ;WAIT
003542' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
003544' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003550' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003554' 012777  000021  176016          MOV     #21,@RMCS1      ;ISSUE A PACK ACK
003562' 032777  000200  176010  1$:     BIT     #BIT7,@RMCS1    ;  FUNCTION DONE ?
003570' 001005                          BNE     2$              ; YES, CONTINUE
003572' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003576' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003602' 000767                          BR      1S              ; NO, WAIT TILL DONE
003604' 012777  177777  176004  2$:     MOV     #-1,@RMAS       ;CLEAR AS BIT
003612' 012777  040000  175760          MOV     #BIT14,@RMCS1   ; CLEAR ANY CONTROLLER ERRORS
003620' 016777  174752  176004          MOV     ADDI,@RMOF      ; SET BIT FOR 11 FORMAT
003626' 000205                          RTS     R5              ; RETURN

003630' 016777  174736  175752  GOGO:   MOV     UNITNO,@RMCS2   ; LOAD UNIT SELECT
003636' 012777  003764' 174144          MOV     #NTRUPT,@VECTOR ; SET INTERRUPT ENTRY POINTER
003644' 032767  001000  174204          BIT     #ADDR22,RES1    ;11/70?
003652' 001434                          BEQ     1S              ;NO
003654' 017767  175724  174622          MOV     @RMBA,PA18      ;GET 18 BIT ADDR
003662' 006267  174620                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
003666' 006267  174614                  ASR     XMEM
003672' 006267  174610                  ASR     XMEM
003676' 006267  174604                  ASR     XMEM
003702' 104416  000000' 000504'         MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
003710' 016777  174574  175666          MOV     PA22,@RMBA      ;LOAD BA REG
003716' 016777  174570  175724          MOV     EA22,@RMBAE     ;LOAD BAE REG
003724' 042767  000034  174560          BIC     #34,EA22        ;CLEAR UNWANTED BITS
003732' 000367  174554                  SWAB    EA22            ;LOAD INTO BITS 8,9
003736' 016767  174550  174542          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
003744' 056767  174536  174522  1S:     BIS     XMEM,FUNC       ;LOAD EXTENDED MEMORY BITS
003752' 016777  174516  175620          MOV     FUNC,@RMCS1     ;EXECUTE THE FUNCTION
003760' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

003764'                         NTRUPT:
                                        ;-----------------------------------------------------------------
003764' 000004  000000' 003772'         PIRQ$,BEGIN,1S          ; QUEUE UP TO CONTINUE AT 1S AND RTI
                                        ;-----------------------------------------------------------------

003772' 004567  000116          1S:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
003776' 000205                          RTS     R5              ; ERRORS DETECTED, RETURN
004000' 005725                          TST     (R5)+           ; NO ERRORS, SKIP RETRY
004002' 005725                          TST     (R5)+
004004' 000205                          RTS     R5              ; RETURN OK
```

```
                                ;
004006' 016700  174446          ROOM:   MOV     BLK1,R0         ; SAVE THE CURRENT BLOCK NUMBER
004012' 016703  174124                  MOV     WBUFSZ,R3       ; GET THE TRANSFER SIZE
004016' 132767  000002  003054          BITB    #BIT1,FLAG      ; PLENTY OF ROOM LEFT ?
004024' 001406                          BEQ     4$              ; YES, CONTINUE
004026' 142767  000001  003044          BICB    #BIT0,FLAG      ; CLEAR 32K INDICATOR
004034' 016701  174464                  MOV     MHICV,R1        ; LOAD MAX. NUMBER OF BLOCKS
004040' 001402                          BEQ     5$              ;EXHAUST OF ALL BLOCKS
004042' 005725                  4$:     TST     (R5)+           ; YES, MUST BE A REAL ERROR
004044' 000205                          RTS     R5              ; RETURN, ERROR
004046' 152767  000004  003024  5$:     BISB    #BIT2,FLAG      ; SET OVERFLOW FLAG
004054' 000205                          RTS     R5              ; RETURN OK
                                ;       ------------------------
                                ;
004056' 012701  000001          DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
004062' 016700  174504                  MOV     UNITNO,R0       ; GET THE DRIVE NUMBER
004066' 001403                          BEQ     2$              ; IF DRIVE 0 GO DROP IT
004070' 006301                  1$:     ASL     R1              ; POINT TO NEXT DRIVE
004072' 005300                          DEC     R0              ; IS THIS THE ONE ?
004074' 001375                          BNE     1$              ; NO, LOOK AGAIN
004076' 040167  174454          2$:     BIC     R1,DVICE        ; DROP THE DRIVE
                                ;       ;*****************************************************************
                                ;                               ;CONVERT UNITNO TO ASCII AND
                                ;                               ;STORE AT ADR1
004102' 104420  000000' 000572'         OTOA$,BEGIN,UNITNO,ADR1
004110' 007054'                 ;       ;*****************************************************************
                                ;
004112' 000207                          RTS     PC              ; RETURN
                                ;       ------------------------
```



```
004114' 005067  174360          ERRORS: CLR     BADSEC          ;CLEAR THE BAD SECTOR FLAG
004120' 005777  175454                  TST     @RMCS1          ; ATTENTION OR ERROR ?
004124' 100402                          BMI     22$             ;YES
004126' 000167  000302                  JMP     RESYNC          ;NO GO ON TO NEXT FUNTION
004132' 032777  001000  175454  22$:     BIT     #BIT9,@RMER1    ; ADDRESS OVERFLOW ?
004140' 001403                          BEQ     1$              ; NO, CONTINUE
004142' 004567  177640                  JSR     R5,ROOM         ; YES, IS IT A REAL ERROR ?
004146' 000532                          BR      RESYNC          ; NO, CONTINUE
004150' 032777  002000  175434  1$:     BIT     #BIT10,@RMDS    ;DID LBT SET?
004156' 001126                          BNE     RESYNC          ;YES
004160' 004767  001430                  JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
004164' 005777  175420                  TST     @RMCS2          ;IS THIS A DATA LATE ERROR?
004170' 100012                          BPL     11$             ;NO
004172' 005267  174256                  INC     DLTCNT          ;ADD 1 TO DATA LATE COUNTER
004176' 032767  000004  173612          BIT     #BIT2,SR1       ;TYPE ERROR AND COUNT IT?
004204' 001102                          BNE     8$              ;NO
004206' 104403  000000' 007010'         MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
004214' 000450                          BR      5$              ;CONT
004216' 032777  020000  175354  11$:     BIT     #BIT13,@RMCS1   ; MASSBUS CONTROL PARITY ERROR ?
004224' 001035                          BNE     3$              ; YES
004226' 032777  000400  175354          BIT     #BIT8,@RMCS2    ; MASSBUS DATA PARITY ERROR ?
004234' 001035                          BNE     4$              ; YES
004236' 032777  040000  175334          BIT     #BIT14,@RMCS1   ; TRANSFER ERROR ?
004244' 001015                          BNE     2$              ; YES
004246' 032777  040000  175336          BIT     #BIT14,@RMDS    ; ANY DRIVE ERRORS ?
004254' 001030                          BNE     5$              ; YES
004256' 005777  175334                  TST     @RMAS           ; ANY ATTENTIONS ACTIVE ?
004262' 001025                          BNE     5$              ; YES, CONTINUE
004264' 005067  173616                  CLR     ERRTYP          ;UNKNOWN ERROR
                                ;       ;*****************************************************************
004270' 104405  000000' 001600'         HRDER$,BEGIN,TABLE      ; SPECIAL CONDITION SET BUT NO REASON FOUND
                                ;       ;*****************************************************************
004276' 000445                          BR      8$              ; RETURN
                                ;
004300' 032777  100000  175334  2$:     BIT     #BIT15,@RMER2   ;A BAD SPOT ? ******
004306' 001044                          BNE     9$              ;BRANCH IF SO *******
004310' 104403  000000' 006752'         MSGN$,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
004316' 000407                          BR      5$              ; GO DUMP REGISTERS
004320'                         3$:
004320' 104403  000000' 006756'         MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
004326' 000403                          BR      5$              ; GO DUMP REGISTERS
004330'                         4$:
004330' 104403  000000' 006762'         MSGN$,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
004336' 005777  175254          5$:     TST     @RMAS           ; ANY ATTENTIONS ACTIVE ?
004342' 001402                          BEQ     6$              ; NO, CONTINUE
004344' 004767  001060                  JSR     PC,WHO          ; YES, FIND OUT WHICH DRIVE IT IS
004350' 016700  175246          6$:     MOV     RMDB,R0         ; SAVE ADDRESS OF DATA BUFFER
004354' 032777  000200  175226          BIT     #BIT7,@RMCS2    ; CAN DATA BUFFER BE READ ?
004362' 001003                          BNE     7$              ; YES, CONTINUE
004364' 012767  000552' 175230          MOV     #ZERO,RMDB      ; NO, LOAD ADDRESS OF ZERO
004372' 012767  000001  173506  7$:     MOV     #1,ERRTYP       ;DATA ERROR
                                ;       ;*****************************************************************
004400' 104406  000000' 001600'         SOFER$,BEGIN,TABLE      ; DUMP RH11 AND RM REGISTERS
                                ;       ;*****************************************************************
004406' 010067  175210                  MOV     R0,RMDB         ; RESTORE DATA BUFFER ADDRESS
004412' 004567  177106          8$:     JSR     R5,CLEAR        ;GO CLEAR OUT ERRORS
```

```
 800  004416' 000205                 RTS     R5              ; ERRORS DETECTED, RETURN
 801  004420' 004567  177100         9$:     JSR     R5,CLEAR        ;CLEAR THE BAD SPOT ERROR
 802  004424' 012767  177777  174046         MOV     #-1,BADSEC      ;SET THE BAD SPOT ERROR FLAG
 803  004432' 000205                 RTS     R5              ;EXIT
```



```
 804  004434' 005725                 RESYNC: TST     (R5)+           ;SKIP RETRY
 805  004436' 000205                         RTS     R5              ; RETURN OK
 806
 807
 808
 809                                 ;THIS ROUTINE DETERMINES IF THERE IS ENOUGH ROOM ON THE DISK TO
 810                                 ;DO ANOTHER TRANSFER. IF NOT, PROGRAM GOES TO RESYNC TO
 811                                 ;ALLOW BOTH PROCESSORS TO RESYNC AND TO RESTART.
 812  004440' 005767  174014         BLOCK:  INC     BLK1            ; STEP TO NEXT BLOCK
 813  004444' 132767  000002  002426         BITB    #BIT1,FLAG      ; BLOCK # IN CYLINDER 410. OR HIGHER ?
 814  004452' 001012                 BNE     1$              ; YES, GO ADJUST BLOCK #
 815  004454' 026767  174042  173776         CMP     HLOWCY,BLK1     ; BLOCK # IN CYLINDER 410. OR LOWER ?
 816  004462' 101017                 BHI     2$              ; YES, RETURN
 817  004464' 005067  173770         CLR     BLK1            ; NO, RESET BLOCK #
 818  004470' 152767  000002  002402         BISB    #BIT1,FLAG      ; SET HIGH RANGE FLAG
 819  004476' 000411                 BR      2$              ; RETURN
 820
 821  004500' 026767  174020  173752 1$:     CMP     MHICY,BLK1      ; BLOCK # IN WITHIN RANGE ?
 822  004506' 101005                 BHI     2$              ; YES, RETURN
 823  004510' 005067  173744         CLR     BLK1            ; NO, RESET BLOCK #
 824  004514' 142767  000002  002355         BICB    #BIT1,FLAG      ; SET FLAG TO LOWER RANGE FOR NEW PACK CYCLE
 825  004522' 016700  173732         2$:     MOV     BLK1,R0         ;TRANSFER PAREMETER FOR CONVRT
 826  004526' 005767  174042                 TST     MIXD#           ;IS AN RM03/RM02 ?
 827  004532' 001432                 BEQ     CONVRT          ;BRANCH IF SO
 828  004534' 016777  174032  175046         MOV     UNITNO,@RMCS2   ;LOAD THE DRIVE ADDRESS
 829  004542' 005067  173714         CLR     CYL             ;RESET CYLINDER ADDRESS
 830  004546' 005067  173712         CLR     CYLSAV          ;RESET THE SAVED CYLINDER ADDRESS
 831  004552' 005067  173776         CLR     DSKADR          ;RESET THE TRACK AND SECTOR ADDRESS
 832  004556' 027777  000620  175050         CMP     #400.,@RMDC     ;OVER CYL 400 ?
 833  004564' 101414                 BLOS    3$              ;BRANCH IF SO
 834  004566' 017767  175042  173666         MOV     @RMDC,CYL       ;
 835                                 ;       MOV     @RMDC,CYLSAV    ;
 836  004574' 062767  000001  173660         ADD     #1,CYL          ;THE NUMBER CAN MODIFY TO INCREMENT
 837                                                                 ;THE CYLINDER ADDRESS
 838  004602' 017767  175000  173744         MOV     @RMDA,DSKADR    ;
 839  004610' 017767  174772  173736         MOV     @RMDA,DSKADR    ;
 840  004616' 000207                 3$:     RTS     PC              ;
```

```
 841  004620' 005067  173636          CONVRT: CLR     CYL             ; CLEAR DISK ADDRESSES
 842  004624' 005067  173636                  CLR     SEC             ;
 843  004630' 105067  002245'                 CLRB    TRK             ;
 844  004634' 016701  173656                  MOV     MBLKTR,R1       ; LOAD REG. 1 WITH BLK PER TRACK
 845                                                                   ; REG 0 HAS BLK1 (BLOCK #) IN IT
 846  004640' 016702  173650                  MOV     MBLKRV,R2       ; LOAD REG. 2 WITH BLK PER REVOLUTION
 847  004644' 132767  000002  002226          BITB    #BIT1,FLAG      ; BLOCK # ON CYLINDER 410. OR LOWER ?
 848  004652' 001403                          BEQ     1$              ; YES, CONTINUE
 849  004654' 012767  000632  173600          MOV     #410.,CYL       ; NO, LOAD HIGH BASE VALUE FOR CYLINDER ADR.
 850  004662' 020001                  1$:     CMP     R0,R1           ; CORRECT CYLINDER FOUND ?
 851  004664' 103413                          BLO     2$              ; YES, CONTINUE
 852  004666' 005267  173570                  INC     CYL             ; NO, STEP TO NEXT CYLINDER
 853  004672' 160100                          SUB     R1,R0           ; SUBTRACT 1 CYLINDER FROM BLOCK #
 854  004674' 000772                          BR      1$              ; CONTINUE UNTIL CORRECT CYLINDER IS FOUND
 855  004676' 022767  001440  173556          CMP     #800.,CYL       ;NOT OVER CYL 800 FOR PROTECT BAD SE.FILE
 856  004704' 101003                          BHI     2$              ;
 857  004706' 012767  000632  173546          MOV     #410.,CYL       ;RESET TO 410
 858  004714' 020002                  2$:     CMP     R0,R2           ; CORRECT TRACK FOUND ?
 859  004716' 002404                          BLT     3$              ; YES, CONTINUE
 860  004720' 105267  002155                  INCB    TRK             ; NO, STEP TO NEXT TRACK
 861  004724' 160200                          SUB     R2,R0           ; SUBTRACT 1 TRACK FROM BLOCK #
 862  004726' 000772                          BR      2$              ; CONTINUE UNTIL CORRECT TRACK IS FOUND
 863  004730' 005700                  3$:     TST     R0              ; CORRECT SECTOR FOUND ?
 864  004732' 001405                          BEQ     4$              ; YES, CONTINUE
 865  004734' 062767  000004  173524          ADD     #4,SEC          ; NO, STEP TO NEXT 1024 WORDS
 866  004742' 005300                          DEC     R0              ; DECREASE BLOCK # BY 1
 867  004744' 000771                          BR      3$              ; CONTINUE UNTIL CORRECT SECTOR IS FOUND
 868  004746' 122767  000005  002125  4$:     CMPB    #5,TRK          ; LAST TRACK ?
 869  004754' 003007                          BGT     5$              ; NO, CONTINUE
 870  004756' 026767  173536  173502          CMP     MODE,SEC        ; YES, LAST SECTOR ?
 871  004764' 003003                          BGT     5$              ; NO, CONTINUE
 872  004766' 162767  000004  173472          SUB     #4,SEC          ; YES, ADJUST SECTOR SO NO OVERFLOW
 873  004774' 016767  173466  173552  5$:     MOV     SEC,DSKADR      ; LOAD SECTOR INTO DISK ADDRESS
 874  005002' 116767  002073  173545          MOVB    TRK,DSKADR+1    ; LOAD TRACK INTO DISK ADDRESS
 875                                                                   ;CKECK THE THE SELECT ADDRESS IS  IN THE BAD SPOT ?
 876  005010' 005000                          CLR     R0              ;INDEX VALUE TO THE BAD SPOT TABLE
 877  005012' 026760  173444  000254' BADLOP: CMP     CYL,BADSPT(R0)  ;CYLINDER ADDRESS IN THE SPOT TABLE?
 878  005020' 001403                          BEQ     1$              ;YES
 879  005022' 062700  000004                  ADD     #4,R0           ;CHECK THE NEXT SPOT
 880  005026' 000434                          BR      5$              ;COMMON BRANCH POINT
 881  005030' 062700  000002          1$:     ADD     #2,R0           ;CHECK THE TRAK AND SEC
 882  005034' 126760  002041  000255' CMPB    TRK,BADSPT+1(R0)        ;ON THE BAD SPOT TRACK ?
 883  005042' 001403                          BEQ     2$              ;YES
 884  005044' 062700  000002                  ADD     #2,R0           ;CHECK THE NEXT SPOT
 885  005050' 000423                          BR      5$              ;COMMON BRANCH POINT
 886  005052' 126760  173410  000254' 2$:     CMPB    SEC,BADSPT(R0)  ;CHECK THE SECTOR ADDRESS IS IN THIS BLOCK RANGE ?
 887  005060' 101015                          BHI     4$              ;NO, NOT IN THIS BLOCK
 888  005062' 062767  000003  173376          ADD     #3,SEC          ;WITHIN THE BLOCK ?
 889  005070' 126760  173372  000254'          CMPB    SEC,BADSPT(R0)  ;NOT IN THE BLOCK IF LESS
 890  005076' 103403                          BLO     3$              ;
 891  005100' 012700  177777                  MOV     #-1,R0          ;ERROR FLAG
 892  005104' 000405                          BR      5$              ;COMMON BRANCH POINT
 893  005106' 162767  000003  173352  3$:     SUB     #3,SEC          ;RESTORE THE SECTOR VALUE
 894  005114' 062700  000002          4$:     ADD     #2,R0           ;CHECK NEXT BAD SPOT
 895  005120' 005700                  5$:     TST     R0              ;MINUS SIGN SET, IF ON A BAD SPOT
 896  005122' 100002                          BPL     .+6             ;SKIP CURRENT SELECTED BLOCK IF BAD SPOT
```



```
 897  005124' 000167  177310                  JMP     BLOCK           ;SELECT OTHER BLOCK
 898  005130' 022700  000200                  CMP     #128.,R0        ;END OF TABLE
 899  005134' 101326                          BHI     BADLOP          ;NO, THEN BRANCH BACK
 900  005136' 000207                          RTS     PC              ; RETURN
 901                                          ;-------------------------
 902
 903  005140' 016767  173316  173316  RM03CK: MOV     CYL,CYLSAV      ;SAVE THE CALCULATED ADDRESS
 904  005146' 005067  173422                  CLR     MIXDV           ;RESET MIX DRIVE FLAG
 905  005152' 022777  024025  174446          CMP     #24025,@RMDT    ;DUAL PORT RM02 ?
 906  005160' 001417                          BEQ     1$              ;BRANCH IF SO
 907  005162' 022777  020025  174436          CMP     #20025,@RMDT    ;SINGLE PORT RM02 ?
 908  005170' 001413                          BEQ     1$              ;BRANCH IF SO
 909  005172' 022777  024024  174426          CMP     #24024,@RMDT    ;DUAL PORT RM03 ?
 910  005200' 001407                          BEQ     1$              ;YES
 911  005202' 022777  020024  174416          CMP     #20024,@RMDT    ;NO...SINGLE PORT RM03 ?
 912  005210' 001403                          BEQ     1$              ;YES
 913                                          ;JSR    R5,DROP         ;DROP THE DRIVE ;IF NOT A RM03
 914                                          ;MESSGN$,DRPL,BEGIN     ;MESSAGE - NOT A RM03
 915                                          ;TST    (SP)+           ;CLEAR STACK FOR WHOMEVER CALLING RM03CK
 916                                          ;MOV    (SP)+,R5        ;CLEAR STACK FOR WHOMEVER CALLING "WRITE",
 917                                          ;                       ;"WRITCK" AND "READ" ROUTINES
 918                                          ;JMP    NEXT            ;TO NEXT DRIVE
 919  005212' 012767  177777  173354          MOV     #-1,MIXDV       ;SET NON-RM03,RM02 DRIVE FLAG
 920  005220' 000207                  1$:     RTS     PC              ;
 921                                          ;-------------------------
 922
```

```
 923  005222' 005267  173344          FDUNIT: INC     UNITNO          ; COUNT A DRIVE
 924  005226' 142767  000010  001644          BICB    #BIT3,FLAG      ; CLEAR END OF DRIVES FLAG
 925  005234' 027767  000010  173330          CMP     #8.,UNITNO      ; ALL DRIVES CHECKED ?
 926  005242' 003404                          BLE     1$              ; YES, GO FLAG END OF DRIVES
 927  005244' 006267  173310                  ASR     DRIVE           ; NO, IS NEXT DRIVE CHOSEN ?
 928  005250' 103364                          BCC     FDUNIT          ; NO, GO TRY ANOTHER DRIVE
 929  005252' 000411                          BR      2$              ; RETURN
 930  005254' 152767  000010  001616  1$:     BISB    #BIT3,FLAG      ; SET END OF DRIVES FLAG
 931  005262' 012767  177777  173302          MOV     #-1,UNITNO      ; RESET DRIVE COUNTER
 932  005270' 016767  173262  173262          MOV     DVICE,DRIVE     ; RESTORE CHOSEN DRIVES
 933  005276' 000207                  2$:     RTS     PC              ; RETURN
 934                                          ;       ------------------------
 935
 936
 937                                          ;SEIZE THE DRIVE BY READING CS1 REG
 938
 939  005300' 012767  077777  173150  NOTRDY: MOV     #77777,CLK      ; SET THE TIMER
 940  005306' 005777  174266          4$:     TST     @RMCS1          ;GRAB THE DRIVE
 941  005312' 032777  004000  174260          BIT     #BIT11,@RMCS1   ;DO I HAVE THE DRIVE DVA?
 942  005320' 001013                          BNE     2$              ;YES
 943  005322' 104407  000000'                BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
 944  005326' 104407  000000'                BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 945  005332' 005367  173120                  DEC     CLK             ;COUNT # OF TRIES
 946  005336' 001363                          BNE     4$              ;NOT DONE YET
 947  005340' 104403  000000' 007014'         MSGNS,BEGIN,NOT ;ASCII MESSAGE CALL WITH COMMON HEADER
 948  005346' 000405                          BR      5$              ;COULD NOT GET DRIVE
 949  005350' 004567  176150          2$:     JSR     R5,CLEAR        ; RESET THE CONTROLLER AND DRIVE
 950  005354' 004567  000122                  JSR     R5,READY        ; IS DRIVE READY ?
 951  005360' 000422                          BR      1$              ; YES, CONTINUE
 952  005362' 004767  000226          5$:     JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
 953  005366' 012767  000006  172512          MOV     #6,ERRTYP       ;DRIVE NOT AVAILABLE
 954                                          ;*********************************************************************
 955  005374' 104405  000000' 001600'         ARDER$,BEGIN,TABLE      ; COULD NOT GET DRIVE....TIME-OUT
 956                                          ;*********************************************************************
 957  005402' 012777  000013  174170          MOV     #13,@RMCS1      ;RELEASE DRIVE
 958  005410' 004767  176442                  JSR     PC,DROP         ; NO, DROP THE DRIVE
 959  005414' 104403  000000' 007024'         MSGNS,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
 960  005422' 000167  175414                  JMP     FINI            ;COULD NOT GET DRIVE DROP MODULE
 961  005426' 000207                  1$:     RTS     PC              ;RETURN
 962                                          ;       ------------------------
```



```
 963  005430' 017701  174162          WHO:    MOV     @RMAS,R1        ; GET THE ATTENTION SUMMARY
 964  005434' 017704  174150                  MOV     @RMCS2,R4       ; SAVE THE STATUS REGISTER
 965  005440' 012702  000001                  MOV     #BIT0,R2        ; SET POINTER TO DRIVE 0
 966  005444' 005003                          CLR     R3              ; ZERO THE DRIVE COUNTER
 967  005446' 030201                  1$:     BIT     R2,R1           ; FIND IT ?
 968  005450' 001006                          BNE     2$              ; YES, CONTINUE
 969  005452' 005203                          INC     R3              ; NO, INCREMENT THE DRIVE COUNTER
 970  005454' 006302                          ASL     R2              ; SET POINTER TO NEXT DRIVE
 971  005456' 032702  000400                  BIT     #BIT8,R2        ; ALL DONE ?
 972  005462' 001771                          BEQ     1$              ; NO, GO AGAIN
 973  005464' 000207                          RTS     PC              ; SOMEBODY LIED -- NO ATTENTIONS SET
 974  005466' 042704  000007          2$:     BIC     #7,R4           ; CLEAR OUT OLD UNIT NUMBER
 975  005472' 050304                          BIS     R3,R4           ; LOAD THE NEW UNIT NUMBER
 976  005474' 010477  174110                  MOV     R4,@RMCS2       ; RESTORE THE STATUS REGISTER
 977  005500' 000207                          RTS     PC              ; RETURN
 978                                          ;       ------------------------
 979
 980  005502' 016777  173064  174100  READY:  MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
 981  005510' 017700  174076                  MOV     @RMDS,R0        ; SAVE STATUS IN R0
 982  005514' 105700                          TSTB    R0              ; DRIVE READY ?
 983  005516' 100022                          BPL     1$              ; NO
 984  005520' 032700  000100                  BIT     #BIT6,R0        ; VOLUME VALID ?
 985  005524' 001417                          BEQ     1$              ; NO
 986  005526' 032700  000400                  BIT     #BIT8,R0        ; DRIVE PRESENT ?
 987  005532' 001414                          BEQ     1$              ; NO
 988  005534' 032700  004000                  BIT     #BIT11,R0       ; WRITE LOCKED ?
 989  005540' 001011                          BNE     1$              ; YES
 990  005542' 032700  010000                  BIT     #BIT12,R0       ; MEDIUM ON LINE ?
 991  005546' 001406                          BEQ     1$              ; NO
 992  005550' 032700  040000                  BIT     #BIT14,R0       ; ANY ERRORS ?
 993  005554' 001003                          BNE     1$              ; YES
 994  005556' 005700                          TST     R0              ; ATTENTION SET ?
 995  005560' 100401                          BMI     1$              ; YES
 996  005562' 000205                          RTS     R5              ; RETURN READY
 997  005564' 005725                  1$:     TST     (R5)+           ; SKIP INSTRUCTION FOLLOWING CALL
 998  005566' 000205                          RTS     R5              ; RETURN AS NOT READY
 999                                          ;       ------------------------
1000
1001
1002  005570' 014167  172312          ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
1003  005574' 010167  172302                  MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
1004  005600' 014267  172304                  MOV     -(R2),AWAS      ; LOAD THE DATA
1005  005604' 010267  172274                  MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
1006  005610' 005721                          TST     (R1)+           ; RESET REG. 1
1007  005612' 005722                          TST     (R2)+           ; RESET REG. 2
1008
1009  005614' 016767  173760  172256  ERSUB1: MOV     RMCS1,CSRA      ; LOAD ADR OF CURRENT CSR
1010  005622' 017767  173752  172252          MOV     @RMCS1,ACSR     ; LOAD CONTENTS OF CURRENT CSR
1011  005630' 000207                          RTS     PC              ; RETURN
```

```
1012
1013
1014
1015
1016
1017  005632' 012777  000040  173750  REZET:  MOV     #BIT5,@RMCS2    ; ISSUE AN RH11 INIT
1018  005640' 012777  177777  173750          MOV     #-1,@RMAS       ;CLEAR ALL ATA BITS
1019  005646' 012767  077777  172602          MOV     #77777,CLK      ; SET THE TIMER
1020  005654' 032777  000200  173716  1$:     BIT     #BIT7,@RMCS1    ; CONTROLLER READY ?
1021  005662' 001017                          BNE     2$              ; YES, CONTINUE
1022  005664' 104407  000000'                 BREAKS,REGIN            ;TEMPORARY RETURN TO MONITOR....
1023  005670' 104407  000000'                 BREAKS,REGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1024  005674' 005367  172556                  DEC     CLK             ; WAIT SOME MORE ?
1025  005700' 001365                          BNE     1$              ; YES
1026  005702' 005067  172650                  CLR     DVICE           ; NO, SET TO DROP THE MODULE
1027  005706' 012767  000004  172172          MOV     #4,ERRTYP       ;CONTROLLER NOT READY
1028                                          ;*********************************************************
1029  005714' 104405  000000' 001600'         ERRORS,REGIN,TABLE      ; CONTROLLER NOT READY
1030                                          ;*********************************************************
1031  005722' 000207                  2$:     RTS     PC              ; RETURN
1032                                  ;       -------------------------
```



```
1033  005724' 016700  172056          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
1034  005730' 010067  173644                  MOV     R0,RMCS1        ; GENERATE REGISTER ADDRESSES
1035  005734' 005720                          TST     (R0)+
1036  005736' 010067  173640                  MOV     R0,RMWC
1037  005742' 005720                          TST     (R0)+
1038  005744' 010067  173634                  MOV     R0,RMBA
1039  005750' 005720                          TST     (R0)+
1040  005752' 010067  173630                  MOV     R0,RMDA
1041  005756' 005720                          TST     (R0)+
1042  005760' 010067  173624                  MOV     R0,RMCS2
1043  005764' 005720                          TST     (R0)+
1044  005766' 010067  173620                  MOV     R0,RMDS
1045  005772' 005720                          TST     (R0)+
1046  005774' 010067  173614                  MOV     R0,RMER1
1047  006000' 005720                          TST     (R0)+
1048  006002' 010067  173610                  MOV     R0,RMAS
1049  006006' 005720                          TST     (R0)+
1050  006010' 010067  173604                  MOV     R0,RMLA
1051  006014' 005720                          TST     (R0)+
1052  006016' 010067  173600                  MOV     R0,RMDB
1053  006022' 005720                          TST     (R0)+
1054  006024' 010067  173574                  MOV     R0,RMMR1
1055  006030' 005720                          TST     (R0)+
1056  006032' 010067  173570                  MOV     R0,RMDT
1057  006036' 005720                          TST     (R0)+
1058  006040' 010067  173564                  MOV     R0,RMSN
1059  006044' 005720                          TST     (R0)+
1060  006046' 010067  173560                  MOV     R0,RMOF
1061  006052' 005720                          TST     (R0)+
1062  006054' 010067  173554                  MOV     R0,RMDC
1063  006060' 005720                          TST     (R0)+
1064  006062' 010067  173550                  MOV     R0,RMHR
1065  006066' 005720                          TST     (R0)+
1066  006070' 010067  173544                  MOV     R0,RMMR2
1067  006074' 005720                          TST     (R0)+
1068  006076' 010067  173540                  MOV     R0,RMER2
1069  006102' 005720                          TST     (R0)+
1070  006104' 010067  173534                  MOV     R0,RMEC1
1071  006110' 005720                          TST     (R0)+
1072  006112' 010067  173530                  MOV     R0,RMEC2
1073  006116' 032767  001000  171732          BIT     #ADDR22,RES1    ;11/70 MONITOR?
1074  006124' 001417                          BEQ     1$              ;NO
1075  006126' 016700  171654                  MOV     ADDR,R0         ;LOCATE THE RMBAE
1076  006132' 062700  000050                  ADD     #50,R0          ;ASSUME 22 REGISTERS
1077  006136' 032767  100000  171652          BIT     #BIT15,SR1      ;32 REGISTER ?
1078  006144' 001402                          BEQ     3$              ;BRANCH IF NOT 32 REGISTERS
1079  006146' 062700  000024                  ADD     #24,R0          ;ADJUST THE RMBAE ADDRESS
1080  006152' 010067  173472          3$:     MOV     R0,RMBAE
1081  006156' 005720                          TST     (R0)+
1082  006160' 010067  173466                  MOV     R0,RMCS3
1083  006164' 016700  171620          1$:     MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1084  006170' 012720  002574'                 MOV     #STRT,(R0)+     ; SET POINTER JUST IN CASE
1085  006174' 116710  171612                  MOVB    BR1,(R0)        ; SET PRIORITY
1086  006200' 000207                  2$:     RTS     PC              ; RETURN
```

```
1087  006202' 020040  051124  047101  MES1:   .ASCIZ  '  TRANSFER  ERROR%'
1088  006210' 043123  051105  020040
1089  006216' 051105  047522  022522
1090  006224'     000
1091  006225'     040  046440  051501  MES2:   .ASCIZ  '  MASSBUS  PARITY  ERROR%'
1092  006232' 041123  051525  020040
1093  006240' 040520  044522  054524
1094  006246' 020040  051105  047522
1095  006254' 022522      000
1096  006257'     040  046440  051501  MES3:   .ASCIZ  '  MASSBUS  DATA  PARITY  ERROR%'
1097  006264' 041123  051525  020040
1098  006272' 040504  040524  020040
1099  006300' 040520  044522  054524
1100  006306' 020040  051105  047522
1101  006314' 022522      000
1102  006317'     040  042040  044522  MES4:   .ASCIZ  '  DRIVE  '
1103  006324' 042526  020040      000  MES5:   .ASCIZ  '  DROPPED%'
1104  006331'     040  042040  047522
1105  006336' 050120  042105  000045  MES6:   .ASCIZ  ' RETRY EXCEEDED%'
1106  006344' 051040  052105  054522
1107  006352' 042440  041530  042505
1108  006360' 042504  022504      000
1109  006365'     040  020040  051127  MES7:   .ASCIZ  '   WRITE'
1110  006372' 052111  000105
1111  006376' 020040  053440  044522  MES8:   .ASCIZ  '   WRITE-CHECK'
1112  006404' 042524  041455  042510
1113  006412' 045503      000
1114  006415'     040  020040  042522  MES9:   .ASCIZ  '   READ'
1115  006422' 042101      000
1116  006425'     040  040504  040524  MES10:  .ASCIZ  ' DATA LATE ERROR%'
1117  006432' 046040  052101  020105
1118  006440' 051105  047522  022522
1119  006446'     000
1120  006447'     040  051104  053111  MES11:  .ASCIZ  ' DRIVE NOT READY%'
1121  006454' 020105  047516  020124
1122  006462' 042522  042101  022531
1123  006470'     000
1124  006471'     040  047503  046125  MES12:  .ASCIZ  ' COULD NOT GET DRIVE%'
1125  006476' 020104  047516  020124
1126  006504' 042507  020124  051104
1127  006512' 053111  022505      000
1128  006517'     137  051537  020137  MES13:  .ASCIZ  '____ NOT A RM03/RM02%'
1129  006524' 047516  020124  020101
1130  006532' 046522  031460  051057
1131  006540' 030116  022462      000
1132  006545'     045  051104  050117  MESLDP: .ASCIZ  '%DROPPED RMDP LOAD DRIVE%'
1133  006552' 042520  020104  046522
1134  006560' 050104  046040  040517
1135  006566' 020104  051104  053111
1136  006574' 022505      000
1137  006577'     045  053117  051105  XMES14: .ASCIZ  '%OVER 32 BAD SECTORS RECORDED,PACK NOT ACCEPTABLE%'
1138  006604' 031440  020062  040502
1139  006612' 020104  042523  052103
1140  006620' 051117  020123  042522
1141  006626' 047503  042122  042105
1142  006634' 050054  041501  020113
```



```
1143  006642' 047516  020124  041501
1144  006650' 042503  052120  041101
1145  006656' 042514  000045
1146  006662' 051045  052105  044522  XMES15: .ASCIZ  '%RETRIEVING THE BAD SPOT FILE FAILS-- DROP MODULE%'
1147  006670' 053105  047111  020107
1148  006676' 044124  020105  040502
1149  006704' 020104  050123  052117
1150  006712' 043040  046111  020105
1151  006720' 040506  046111  026523
1152  006726' 020055  051104  050117
1153  006734' 046440  042117  046125
1154  006742' 022505      000
1155                  006746'                 .EVEN
1156  006746' 006545'                 LDRIVE: MESLDP
1157  006750' 177777                         -1
1158  006752' 006202'                 TRERR:  MES1
1159  006754' 177777                         177777
1160  006756' 006225'                 MCPERR: MES2
1161  006760' 177777                         177777
1162  006762' 006257'                 MDPERR: MES3
1163  006764' 177777                         177777
1164  006766' 006365'                 EXCED1: MES7
1165  006770' 006344'                         MES6
1166  006772' 177777                         177777
1167  006774' 006376'                 EXCED2: MES8
1168  006776' 006344'                         MES6
1169  007000' 177777                         177777
1170  007002' 006415'                 EXCED3: MES9
1171  007004' 006344'                         MES6
1172  007006' 177777                         177777
1173  007010' 006425'                 DLTERR: MES10
1174  007012' 177777                         177777
1175
1176  007014' 006447'                 NOT:    MES11
1177  007016' 177777                         177777
1178  007020' 006471'                 TOUT:   MES12
1179  007022' 177777                         177777
1180
1181  007024' 006317'                 DRP:    MES4
1182  007026' 007061'                         NUMB
1183  007030' 006331'                         MES5
1184  007032' 177777                         177777
1185  007034' 006317'                 DRP1:   MES4
1186  007036' 007061'                         NUMB
1187  007040' 006517'                         MES13
1188  007042' 177777                         177777
1189  007044' 006577'                 MES14:  XMES14
1190  007046' 177777                         177777
1191  007050' 006662'                 MES15:  XMES15
1192  007052' 177777                         177777
1193
1194  007054' 000005                  ADR1:   .BLKB   5
1195  007061'     000                 NUMB:   .BYTE   0
1196  007062'     000                         .BYTE   0
1197                  007064'                 .EVEN
1198  007064'     000                 TRY1:   .BYTE   0
```

```
 1199  007065*     000                TRY2:   .BYTE   0
 1200  007066*     000                TRY3:   .BYTE   0
 1201  007067*     000                TRY4:   .BYTE   0
 1202  007070*     000                TRY5:   .BYTE   0
 1203  007071*     000                TRY6:   .BYTE   0
 1204  007072*     000                TRY7:   .BYTE   0
 1205  007073*     000                TRY8:   .BYTE   0
 1206  007074*     000                TRY9:   .BYTE   0
 1207  007075*     000                TRY10:  .BYTE   0
 1208  007076*     000                TRY11:  .BYTE   0
 1209  007077*     000                TRY12:  .BYTE   0                ;NEED DO NOT DELETE FROM CODE
 1210  007100*     000                FLAG:   .BYTE   0
 1211  007101*     000                TRK:    .BYTE   0
 1212  007102*     000                FIX:    .BYTE   0
 1213          007104*                        .EVEN
 1214                                ;       ------------------------
 1215
 1216          000001                        .END
```


CROSS REFERENCE TABLE -- USER SYMBOLS

```
ABW     002664R          539    542#
ACSR    000102R          219#  1010#
ADDR    000006R          185#  1033    1075
ADDR22= 001000           248#   689    1073
ADR1    007054R          139   1194#
ASB     000106R          223#  1002#
ASTAT   000104R          221#
AWAS    000110R          224#  1004*
BADFL   002212R          461#   505
BADFL2  002270R          470#   515
BADLOP  005012R          877#   899
BADSEC  000500R          326#   576    578*   744*   802*
BADSPT  000254R          248    249#   456    480    877    882    886    889
BEGIN   000000R          182#   431    471    499    516    524    529    565    569    573    585    592    599
                         605    674    675    679    680    696    704    708    739    760    774    780    783
                         796    943    944    947    955    959   1022   1023   1029
BIT0    = 000001         248#   602    720    965
BIT1    = 000002         248#   718    813    818    824    847
BIT10   = 002000         248#   752
BIT11   = 004000         248#   941    988
BIT12   = 010000         248#   405    990
BIT13   = 020000         248#   762
BIT14   = 040000         248#   683    766    768    992
BIT15   = 100000         248#   778   1077
BIT2    = 000004         248#   556    558    725    758
BIT3    = 000010         248#   445    502    538    924    930
BIT4    = 000020         248#
BIT5    = 000040         248#   407   1017
BIT6    = 000100         248#   984
BIT7    = 000200         248#   677    791   1020
BIT8    = 000400         248#   764    971    986
BIT9    = 001000         248#   748
BLKSAV  000562R          366#   527*
BLK1    000460R          318#   440*   527    559*   716    812*   815    817*   821    823*   825
BLOCK   004440R          652    842#   897
BRFARS= 104407           248#   674    675    679    680    943    944   1022   1023
BR1     000012R          197#  1085
BR2     000013R          188#
BTODS$  = 104421         248#
BUFIN   000600R          230    373#   479
CDATAS= 104412           248#   565
CDERCT  000144R          238#
CDMDCT  000146R          239#
CLEAR   003524R          270#   799    801    949
CLK     000456R          317#   939*   945*  1019*  1024*
CNT     000470R          322#   398    403*   520
CNT1    000472R          323#   518    568*
CONFIG  000056R          207#
CONVRT  004620R          827    841#
CSRA    000100R          217#  1009*
CT      002160R          449    451#
CYCLE   003026R          567#
CYL     000462R          319#   461*   627    628*   642    643*   660    661*   829*   834*   836*   841*   849*
                         852*   855    857*   877    903
CYLSAV  000464R          320#   628    643    661    830*   903*
DAOCL   002756R          558#
```

```
DADST   002070R          435     438#
DATCK$= 104411           248#
DATER$= 104404           248#
DLTCNT  000454R          316     404*    757*
DLTERR  007010R          760    1173#
DRIVE   000560R          365#    436*    927*    932*
DROP    004056R          604     729#    958
DRP     007024R          605     959    1181#
DRP1    007034R         1185#
DSKADR  000554R          363#    462*    463*    466*    512*    513*    514     625     640     658     831*    838*    839*
                         873#    874*
DVICE   000556R          364#    418*    428     430*    434     438     452     535     735*    932    1026*
DVIDI   000514R          365#    419
EA22    000512R          335#    698     699*    700*    701
ENDIT$= 104413           248#    569
END$    = 104410         248#    573
ERRORS  004114R          711     744#
ERRTYP  000106R          222#    772#    794*    953*   1027*
ERSUB1  005614R          754     952    1009#
ERSUB2  005570R         1002#
EXCED1  006766R          585    1164#
EXCED2  006774R          592    1167#
EXCED3  007002R          599    1170#
EXIT$   = 104400         248#    704
FDUNIT  005222R          444     501     532     923#    928
FERADR  000502R          327#    397
FILE    000252R          248#
FINT    003042R          436     447     455     500     517     537     572#    960
FIX     007102R         1212#
FLAG    007100R          417#    445     502     538     556     558*    718     720*    725*    813     818*    824*    847
                         924*    930*   1210#
FREE    000150R          240#
FUNC    000474R          324#    622*    637*    655*    702*    703
GETPAS= 104415           248#    441     524
GOA     002776R          557     561#    590
GOB     003006R          563#    597
GOC     002736R          554#    583
GOGO    003630R          636     651     669     687#
GWBUF$= 104414           248#    529
HRDCNT  000044R          202#
HRDER$= 104405           248#    774     955    1029
HRDPAS  000050R          204#
ICONT   000036R          199#
ICOUNT  000040R          200#
IDNUM   000122R          229#
IMODX.= 000000           241#    530
INIT    000030R          196#
INTR    000120R          228#    402*
LDRIVE  006746R          431    1156#
MAP22$= 104416           248#    696
MBLKRV  000514R          336#    411     846
MBLKTR  000516R          337#    844
MCPERR  006756R          783    1160#
MDPERR  006762R          796    1162#
MESLDP  006545R         1132#   1156
MESI    006202R         1087#   1158
```



```
MES10   006425R         1116#   1173
MES11   006447R         1120#   1176
MES12   006471R         1124#   1179
MES13   006517R         1128#   1187
MES14   007044R          499    1189#
MES15   007050R          516    1191#
MES2    006225R         1091#   1160
MES3    006257R         1096#   1162
MES4    006317R         1102#   1181    1185
MES5    006331R         1104#   1183
MES6    006344R         1106#   1165    1168    1171
MES7    006365R         1109#   1164
MES8    006376R         1111#   1167
MES9    006415R         1114#   1170
MHICY   000524R          340#    721     821
MIXDV   000574R          371#    473     826     904*    919*
MLOWCV  000522R          339#    815
MODF    000520R          338#    870
MODNAM  000000R          183#
MODSP   000252R          197     246#
MOD1    000576R          372#    405*    409*    464     684
MSGN$   = 104403         248#    431     499     516     585     592     599     605     760     780     783     786     947
                         959
MSGS$   = 104402         248#
MSG$    = 104401         248#
NEXT    002620R          532#    571     575     606
NEXTA   003206R          579     586     593     600     602#
NOT     007014R          947    1176#
NOTRDY  005300R          450     506     545     939#
NTRUPT  003764R          688     706#
NOLI    = 000000         248#
NUMB    007061R         1182    1186    1195#
NXT1    003046R          575#
ONCEE   000476R          325#    416*
OPEN    = 000000         184     190     191     192     193     210     211     212     213     214     215     216     217
                         219     221     223     224     226     227     228     231     232     234     235     237     238
                         239     240     248#
OTOA$   = 104420         248#    739
PASCNT  000034R          198#
PA18    000504R          332#    691*    696
PA22    000510R          334#    697
PIRQ$   = 000004         248#    708
POPSP   = 005726         248#
POPSP2= 022626           248#
PRTV    = 000000         248#
PRTV0   = 000000         188     248#
PRTV1   = 000040         248#
PRTV2   = 000100         248#
PRTV3   = 000140         248#
PRTV4   = 000200         248#
PRTV5   = 000240         187     248#
PRTV6   = 000300         248#
PRTV7   = 000340         248#
PS      = 177776         248#
PSW     = 177776         248#
PUSH    = 005746         248#
```

```
PUSH2 = 024646          248#
RANDS = 104417          249#
RANNUM  000054R         206#
RBUFEA  000130R         232#    663
RBUFPA  000126R         231#    565     657
RBUFS7  000132R         233#    475     525
RBUFVA  000124R         230#    471     524
READ    003432R         477     563     655#
READY   005502R         448     504     543     950     980#
RESTRT  002534R         225     518#
RESYNC  004434R         447     751     753     804#
RES1    000056R         208#    689    1073
RES2    000060R         209#
RETRX   002502R         478     512#
RETRY1  003052R         555     576#
RETRY2  003154R         562     587#
RETRY3  003154R         564     594#
REZET   005632R         443    1017#
RMAS    001616R         382#    682*    770     787     963    1018*   1048*
RMBA    001604R         377#    624*    639*    657*    691     697*   1038*
RMBAE   001650R         395#    698*   1080*
RMCS1   001600R         375#    671*    676*    677     683*    703*    745     762     766     940     941     957*    1009
                       1010    1020    1034*
RMCS2   001610R         379#    467*    533*    670*    687*    755     764     791     828*    964     976*    980*    1017*
                       1042#
RMCS3   001652R         396#   1082*
RMDA    001606R         378#    469*    625*    640*    658*    838     839    1040*
RMDB    001622R         384#    790     793*    798*   1052*
RMDC    001634R         389#    468*    627*    642*    660*    832     834    1062*
RMDS    001612R         380#    752     768     981    1044*
RMDT    001626R         386#    905     907     909     911    1056*
RMEC1   001644R         393#   1070*
RMEC2   001646R         394#   1072*
RMER1   001614R         381#    748    1046*
RMER2   001642R         392#    778    1068*
RMHR    001636R         390#   1064*
RMLA    001620R         383#   1050*
RMMR1   001624R         385#   1054*
RMMR2   001640R         391#   1066*
RMOF    001632R         388#    684*   1060*
RMSN    001630R         387#   1058*
RMWC    001602R         376#    623*    638*    656*   1036*
RMO3CK  005140R         472     542     626     641     659     903#
ROOM    004006R         716#    750
RSTPT   000112R         225#
RSTRT1  002554R         510     519     521     523#
SBADR   000102R         218#   1003*
SEC     000466R         321#    842*    865*    870     872*    873     886     888*    889     893*
SETUP   005724R         442    1033#
SOFCNT  000042R         201#
SOFERS= 104406          248#    796
SOFPAS  000046R         203#
SPOINT  000032R         197#    439
SPSIZ = 000040          198#    441
SR1     000016R         190#    407     602     758    1077
SR2     000020R         191#
```



```
SR3     000022R         192#
SR4     000024R         193#
START   001662R         196     400#    522
STAT    000026R         194#
STRT    002574R         527#    541     560    1084
STRT1   002602R         528#
SVR0    000062R         210#
SVR1    000064R         211#
SVR2    000066R         212#
SVR3    000070R         213#
SVR4    000072R         214#
SVR5    000074R         215#
SVR6    000076R         216#
SVSCNT  000052R         205#
TABLE   001600R         374#    774     796     955    1029
TBUF    000564R         367#
TOUT    007020R        1178#
TRERR   006752R         790    1158#
TRK     007101R         843*    860*    868     874     882    1211#
TRPDFD= 000022          248#
TRY1    007064R         546*    580*    581    1198#
TRY10   007075R        1207#
TRY11   007076R         551*   1208#
TRY12   007077R        1209#
TRY2    007065R         547*    588    1199#
TRY3    007066R         547*    594*    595    1200#
TRY4    007067R        1201#
TRY5    007070R         548*   1202#
TRY6    007071R        1203#
TRY7    007072R         549*   1204#
TRY8    007073R        1205#
TRY9    007074R         550*   1206#
T16     000526R         343#    406
T18     000540R         352#    410
UNITNO  000572R         370#    441*    467     508*    533     540*    670     687     730     739     828     923*    925
                        931#    980
UPDATT  003424R         552     652#
VECTOR  000010R         186#    688*   1083
WASADR  000104R         220#   1005*
WBUFEA  000136R         235#    630     645
WBUFPA  000134R         234#    624     639
WBUFRQ  000140R         236#
WBUFS7  000142R         237#    530     717
WCNT1   000566R         368#    530*    531*    623     638
WCNT2   000570R         369#    475*    476*    525*    526*    656
WDFR    000116R         227#    400*
WDTO    000114R         226#    401*
WHO     005430R         789     963#
WRITCK  003330R         561     637#
WRITE   003734R         554     622#
XFERAD  001654R         397#
XFERCT  001656R         398#
XFLAG   000005R         184#
XMEM    000506R         333#    635*    650*    668*    692*    693*    694*    695*    701*    702
XMES14  006577R        1137#   1189
XMES15  006662R        1146#   1191
```

```
ZERO    000552R          362#    793
.     = 007104R          373#    446     465     566     577     896    1155#   1194#   1197#   1213#

. ABS.  000000     000
        007104     001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRMAD0,XRMAD0/SOL/CRF:SYM=DDXCOM,XRMAD0
 RUN-TIME:  2  3  .5 SECONDS
 RUN-TIME RATIO: 20/6=4.5
 CORE USED:  7K  (13 PAGES)
```

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E968C-MC

PRODUCT NAME: CXRMBC0 RH11/RM03 DUAL PORT MOD

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RMB IS AN IOMOD THAT EXERCISES RM03/RM02 DISK DRIVES ON AN RH11 CONTROLLER. IT EXERCISES THE DRIVES BY DOING WRITES, WRITE-CHECKS, READS, AND IN-CORE COMPARISONS. ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE TTY.

2. REQUIREMENTS

HARDWARE: 1 TO 8 RM03/RM02'S WITH TWO RH11 CONTROLLERS

STORAGE:: RMB REQUIRES:

1. DECIMAL WORDS: 1793
2. OCTAL WORDS: 03401
3. OCTAL BYTES: 7002

3. PASS DEFINITION

ONE PASS OF THE RMB MODULE CONSISTS OF 300 CYCLES OF THE BASIC TEST SEQUENCE. A-PORT DOES A WRITE, WRITE-CHECK,READ,DATA COMPARE ON THE CURRENT SECTOR (CALLED BLK THROUGHOUT PROGRAM) AND THEN WRITES BLK 0 WITH BIT 4 SET IN THE FLAG WORD (THE FIRST WORD OF BLOCK ZERO), AND THE SECOND WORD WITH THE CURRENT BLK. A-PORT THEN SITS IN A LOOP, PERIODICALLY READING BLK 0 TO SEE IF B-PORT HAS MODIFIED THE FLAG.
B-PORT SITS IN A LOOP WAITING FOR THE FLAG IN BLK ZERO TO BE WRITTEN WITH BIT4 SET, WHEN IT IS B-PORT PICKS THE CURRENT BLK FROM THIS DATA READ FROM BLK 0 AND GOES TO THAT SECTOR AND READS THE DATA THAT A-PORT JUST WROTE. THEN B-PORT WRITES THE DATA BACK AND WRITE CHECKS IT. NEXT IT RE-WRITES BLK 0 WITH BIT4 CLEARED AND WITH BIT2 SET, INDICATING TO A-PORT THAT HE IS DONE. A-PORT AFTER READING BLK 0 AND SEEING THE FLAGS REVERSED, RE-READS THE DATA WHICH IT HAD WRITTEN AND AGAIN DOES AN IN-CORE COMPARE. THIS VERIFIES THE ABILITY OF A-PORT TO RE-READ DATA WHICH IT HAD ORIGINALLY WROTE BUT THAT HAD BEEN READ AND RE-WRITTEN BY B-PORT. B-PORT DOES NOT DO ANY IN CORE COMPARISONS.

B PORT SHOULD NEVER ENCOUNTER A BAD BLK BECAUSE IT ONLY USES BLK'S WHICH A-PORT HAS ALREADY SUCCESSFULLY USED. THEREFORE, WHEN A-PORT GETS 4 ERRORS WHILE ATTEMPTING A SEQUENCE ON ONE BLK, IT MOVES TO THE NEXT BLK, NEVER LETTING B-PORT EVEN TRY THIS BAD BLK. IF B-PORT GETS 4 ERRORS ON A BLK, THAT DRIVE IS DROPPED BECAUSE IT IS MOST LIKELY FAULTY SINCE A-PORT WAS JUST ABLE TO SUCCESSFULLY USE THIS BLK.

LOCATION 206 THROUGH 244 CONTAIN ROOM FOR 16 BAD BLKS. ON ERROR TYPEOUTS WHICH DUMP THE RH REGISTERS, THE LAST ITEM TYPED IN THE TABLE IS THE CURRENT BLK NUMBER. ENTER THIS INTO THE BAD BLK TABLE TO AVOID ERRORS FROM KNOWN MEDIA BAD SPOTS. THIS TABLE ONLY MAKES SENSE ON THE A-PORT SIDE SINCE B-PORT

ALWAYS GETS ITS BLK ADDRESSES FROM A-PORT. IF YOU MODIFY
THE WRITE BUFFER SIZE, YOU MUST ADD SECTORS TO THE BAD BLK
TABLE TO AVOID THE ERRORS. FOR EXAMPLE, IF BLK 3474 IS BAD
AND YOU DOUBLE THE WRITE TRANSFER SIZE TO 1000 OCTAL BYTES
YOU MUST ADD BLK 3473 TO THE TABLE SO THE PROGRAM DOES NOT START
A TRANSFER THAT WILL EXTEND ON INTO THE KNOWN BAD BLK.

B-PORT MUST HAVE SR1 BIT 4 SET. ***** SR1 = 20 (OCTAL) *****
THIS MODULE IS A DUAL-PORT
MODULE TEST, AND ONLY USES THE FIRST 77777 BLKS
OF THE PACK. IF YOU WANT TO VERIFY THE COMPLETE MEDIUM, YOU
SHOULD RUN RMA MODULE, THE NORMAL SINGLE PORT TEST.

4. EXECUTION TIME

ONE PASS OF RMB RUNNING ALONE ON A PDP-11/70 TAKES APPROXIMATELY ONE MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 176700, VECTOR: 254, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SETUP

MAKE CERTAIN THAT ALL DRIVES ARE POWERED UP, WRITE ENABLED, AND READY

THIS MODULE ALSO SUPPORTS RP04/5/6 ON THE SAME MASSBUS CONTROLLER.
HOWEVER,THIS MODULE IS NOT USED TO EXERCISE RP04/5/6 ALONE.

7. OPERATION OPTIONS

SR1 BIT2 SET(1):
COUNT DATA LATE ERRORS BUT DO NOT TYPE THEM OUT

SR1 BIT2 CLEAR(0):
TYPE OUT DATA LATE ERRORS AND COUNT THEM

SR1 BIT4 SET (1) ;-B-PORT PROGRAM *****SR1 = 20 (OCTAL) *****

SR1 BIT15 SET (1):
32 REGISTER OPTION ON RH70

SR1 BIT15 CLEAR (0):
22 REGISTER OPTION ON RH70

8. NON-STANDARD PRINTOUTS

A. MOST PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT

B. ERROR MESSAGES DUMP THE CONTENTS OF THE 20 RH11/RM03 REGISTERS IN THE FOLLOWING ORDER:

RMCS1	RMWC	RMBA	RMDA	RMCS2	RMDS	RMER1	RMAS
RMLA	RMDB	RMMR1	RMDT	RMSN	RMOF	RMDC	RMHR
RMMR2	RMER2	RMEC1	RMEC2	BLK			

9. DUAL PORT SETUP:

TO RUN A DUAL PORT SYSTEM, SR1 HAS TO BE MODIFIED TO INDICATE TO THE MODULE, WHICH PORT THE MODULE IS LOCATED ON. SEE SECTION 7. FOR SR1 OPTIONS.

THE CONTROLLER SELECT SWITCH ON THE RM03 MUST BE IN THE A/B POSITION. THIS SWITCH IS ACTIVATED WHEN THE DRIVE IS CYCLED UP. IF SWITCH WAS NOT IN THIS POSITION WHEN DRIVE WAS POWERED UP, THE FOLLOWING STEPS MUST BE TAKEN. PLACE THE SWITCH IN THE A/B POSITION, DISABLE THE DRIVE, (USING THE DISABLE SWITCH ON THE DRIVE), THEN ENABLE THE DRIVE WITH THE SAME SWITCH. THIS WILL PUT THE RM03 IN THE DUAL PORT MODE.

10. BAD SPOT FILE

TOTAL 16 BAD BLOCKS CAN BE RETRIEVED FROM PORT A.
THE TABLE LABELED "BADSPT" IS SET UP TO RETRIEVE ALL BAD SPOT FILES FROM ALL DRIVES ASSIGNED IN THE BIT MAP "DVID1".
(CYLINDER 822, TRACK 4, SECTOR 0 - MANUFACTUR BAD SPOT FILE)
(CYLINDER 822, TRACK 4, SECTOR 12 - USER BAD SPOT FILE)
LOCATION 1722 MUST BE SET TO 10 TO SEARCH USER BAD SPOT FILE


```
000000'          IDMODX  <RMBC  >,176700,254,5,0,0,300,145,RBUF,256.,256.
000000'          MODULE  150000,RMBC  ,176700,254,5,0,0,300,145,RBUF,256.,256.
                 .TITLE  RMBC DEC/X11 SYSTEM EXERCISER MODULE
                 ;       DDXCOM  VERSION  6       23-MAY-78
                                  .LIST   BIN
                                 ;*****************************************************************************
000000'                          BEGIN:
000000' 046522  041502     040   MODNAM: .ASCII  /RMBC / ;MODULE NAME.
000005'    000                   XFLAG:  .BYTE   0PEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 176700                   ADDR:   176700+0                ;1ST DEVICE ADDR.
000010' 000254                   VECTOR: 254+0                   ;1ST DEVICE VECTOR.
000012'    240                   BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013'    000                   BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                   DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                   SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                   SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                   SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                   SR4:    OPEN                    ;SWITCH REGISTER 4
                                 ;*****************************************************************************
000026' 150000                   STAT:   150000                  ;STATUS WORD.
000030' 001604'                  INIT:   START                   ;MODULE START ADDR.
000032' 000252'                  SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                   PASCNT: 0                       ;PASS COUNTER.
000036' 000300                   ICONT:  300                     ;# OF ITERATIONS PER PASS=300
000040' 000000                   ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                   SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                   HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                   SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                   HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                   SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                   RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                          CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                   RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                   RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                   SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                   SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                   SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                   SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                   SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                   SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                   SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                   CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                          SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                   ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                          WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                   ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                          ERRTYP:                         ;TYPE OF ERROR
000106' 000000                   ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                   AWAS:   OPEN                    ;ACTUAL DATA.
000112' 002236'                  RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                   WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                   WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                   INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000145                   IDNUM:  145                     ;MODULE IDENTIFICATION NUMBER=145
000124' 000444'                  RBUFVA: RBUF                    ;READ BUFFER VIRTUAL ADDRESS
000126' 000000                   RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
```

```
        000130' 000000                  RBUFEA: OPEN                    ;READ BUFFER EA BITS
        000132' 000400                  RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
        000134' 000000                  WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
        000136' 000000                  WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
        000140' 000400                  WBUFRQ: 256.                    ;WRITE BUFFER SIZE REQUESTED
        000142' 000000                  WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
        000144' 000000                  CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
        000146' 000000                  CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
        000150' 000000                  FREE:   OPEN                    ;RESERVED FOR FUTURE USE
                000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                                .NLIST
                                                .WORD   0
                                                .LIST
                                                .ENDR
        000252'                         MODSP:
                                        ;*****************************************************************************
   270  000252' 000000                  BADSPT: 0               ;BAD SPOT TABLE
   271  000254' 000000                          0               ;
   272  000256' 000000                          0               ;.
   273  000260' 000000                          0               ;
   274  000262' 000000                          0               ;BLK 0 IS NEVER CONSIDERED BAD
   275  000264' 000000                          0               ;IF YOURS IS, YOU CANNOT RUN THIS PROGRAM
   276  000266' 000000                          0               ;GO FIND ANOTHER PACK AND TRY AGAIN
   277  000270' 000000                          0
   278  000272' 000000                          0
   279  000274' 000000                          0
   280  000276' 000000                          0
   281  000300' 000000                          0
   282  000302' 000000                          0
   283  000304' 000000                          0
   284  000306' 000000                          0
   285  000310' 000000                          0
   286  000312' 000000  000000  000000  S:      0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0
   287  000320' 000000  000000  000000
   288  000326' 000000  000000  000000
   289  000334' 000000  000000  000000
   290  000342' 000000  000000  000000
   291  000350' 000000  000000  000000
   292  000356' 000000  000000  000000
   293  000364' 000000
   294                                  ;DO NOT CHANGE THE ORDER OF THE NEXT 4 LOCATIONS
   295                                  ;NEEDED FOR MAP22 ROUTINE
   296  000366' 000000                  PA18:   0
   297  000370' 000000                  XMEM:   0
   298  000372' 000000                  PA22:   0
   299  000374' 000000                  EA22:   0
   300  000376' 000000                  BLK:    0
   301  000400' 000000                  DSKADR: 0
   302  000402' 000000                  CYL:    0
   303  000404' 000000                  TRACK:  0
   304  000406' 000000                  FLAG:   0
   305  000410' 000000                  DVICE:  0
   306  000412' 000000                  DRIVE:  0
   307  000414' 000000                  UNITNO: 0
   308  000416' 000000                  CNT:    0
   309  000420' 000000                  CNT1:   0
```



```
   310  000422' 000000                  DLTCNT: 0
   311  000424' 000000                  FUNC:   0
   312  000426' 000000                  TIMER:  0
   313  000430' 000000                  ZERO:   0
   314  000432' 000000                  FERADR: 0
   315  000434' 000000                  CLK:    0
   316  000436' 000000                  TRY:    0
   317  000440' 000000                  M1XDV:  0       ;M1X DRIVE FLAG,=-1,IF NOT RM03,RM02
   318  000442' 000000                  BADSEC: 0               ;HARDWARE DETECT BAD SECTOR
   319  000444' 000400                  RBUF:   .BLKW   256.
   320
   321  001444'                         TABLE:
   322  001444' 000312'                         S
   323  001446' 000314'                         S+2
   324  001450' 000316'                         S+4
   325  001452' 000320'                         S+6
   326  001454' 000322'                         S+10
   327  001456' 000324'                         S+12
   328  001460' 000326'                         S+14
   329  001462' 000330'                         S+16
   330  001464' 000332'                         S+20
   331  001466' 000334'                         S+22
   332  001470' 000336'                         S+24
   333  001472' 000340'                         S+26
   334  001474' 000342'                         S+30
   335  001476' 000344'                         S+32
   336  001500' 000346'                         S+34
   337  001502' 000350'                         S+36
   338  001504' 000352'                         S+40
   339  001506' 000354'                         S+42
   340  001510' 000356'                         S+44
   341  001512' 000360'                         S+46
   342  001514' 000376'                         BLK
   343  001516' 177777                          177777
   344  001520' 000000                  RMCS1:  0
   345  001522' 000000                  RMWC:   0
   346  001524' 000000                  RMBA:   0
   347  001526' 000000                  RMDA:   0
   348  001530' 000000                  RMCS2:  0
   349  001532' 000000                  RMDS:   0
   350  001534' 000000                  RMER1:  0
   351  001536' 000000                  RMAS:   0
   352  001540' 000000                  RMLA:   0
   353  001542' 000000                  RMDB:   0
   354  001544' 000000                  RMMR1:  0
   355  001546' 000000                  RMDT:   0
   356  001550' 000000                  RMSN:   0
   357  001552' 000000                  RMOF:   0
   358  001554' 000000                  RMDC:   0
   359  001556' 000000                  RMHR:   0
   360  001560' 000000                  RMMR2:  0
   361  001562' 000000                  RMER2:  0
   362  001564' 000000                  RMEC1:  0
   363  001566' 000000                  RMEC2:  0
   364  001570' 000000                  RMBAE:  0
   365  001572' 000000                  RMCS3:  0
```

```
366  001574' 000432'                XFERAD: FERADR
367  001576' 000416'                XFERCI: CNT
368  001600' 000376'                XBLK:   BLK
369  001602' 177777                         177777
```



```
370  001604' 012767  000400  176302  START:  MOV     #256.,WD1O      ;256 WORDS TO MEM/ITERATION
371  001612' 012767  000400  176276          MOV     #256.,WDFR      ;256 WORDS FROM MEM/ITERATION
372  001620' 012767  000003  176272          MOV     #3,INTR         ;3 INTERRUPTS/ITERATION
373  001626' 016767  176162  176554          MOV     DVID1,DVICE     ;GET DRIVES TO TEST
374  001634' 016706  176172                  MOV     SPOINT,R6       ;INIATE STACK POINT
375  001640' 012767  000001  176530          MOV     #1,BLK          ;START AT BLOCK(SECTOR) 1
376  001646' 012767  000001  176532          MOV     #1,FLAG         ;SET 1ST TIME FLAG BIT
377  001654' 012767  177777  176532          MOV     #-1,UNITNO      ;INITIATE UNIT NUMBER
378  001662' 004767  004102                  JSR     PC,SETUP
379  001666' 004767  002542                  JSR     PC,PICKDR       ;INITIATE UUT
380  001672' 000240                          NOP                     ;DUMMY ERROR RETURN
381  001674' 004767  004332                  JSR     PC,SETUP2       ;SETUP PSEL BIT
382  001700' 004767  003744                  JSR     PC,REZET        ;CLEAR THE RH
383  001704' 104415  000000' 000124'         GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
384  001712' 005067  176500                  CLR     CNT             ;RESET END-OF-PASS COUNTER
385  001716' 032767  000020  176072          BIT     #20,SR1         ;8-PORT 7,
386  001724' 001161                          BNE     LOOP1           ;THEN,DON'T HAVE TO RETRIEVE THE BAD SPOT FILE
387  001726' 012767  177777  176460          MOV     #-1,UNITNO      ;RESET UNIT NUMBER
388  001734' 012701  000252'                 MOV     #BADSPT,R1      ;TABLE ENTRY
389  001740' 012702  000020                  MOV     #16.,R2         ;16 WORDS
390  001744' 005021                  1$:     CLR     (R1)+           ;CLEAR THE TABLE
391  001746' 005302                          DEC     R2              ;DECREMENT ONE WORD
392  001750' 001375                          BNE     1$              ;BRANCH IF NOT DONE
393  001752' 004767  002456          RES2X:  JSR     PC,PICKDR       ;PICK UP THE FIRST ON LINE DRIVE
394  001756' 000520                          BR      10$             ;BRANCH IF ALL DRIVES ARE ACCESSED
395  001760' 005067  176452                  CLR     TRY             ;CLEAR RETRY COUNTER
396  001764' 012767  000770  176414          MOV     #770,FLAG       ;SPECIAL FLAG TAG FOR RETRIEVE BAD SPOT FILE
397  001772' 012767  001466  176402          MOV     #822.,CYL       ;CYL 822
398  002000' 112767  000004  176373          MOVB    #4,DSKADR+1     ;TRACK 4
399  002006' 112767  000000  176364          MOVB    #0,DSKADR       ;SEC 0 (SEC 12 USER)
400  002014' 004567  003440                  JSR     R5,READY        ;ACCESS THE DRIVE
401  002020' 000402                          BR      .+6             ;BRANCH IF READY
402  002022' 004767  003174                  JSR     PC,NOTRDY       ;WAIT IF NOT READY
403  002026' 005767  176406                  TST     MIXDV           ;EXIT,IF NOT RM03 OR RM02
404  002032' 100471                          BMI     9$              ;EXIT
405  002034' 004567  001222                  JSR     R5,READ         ;READ THE BAD SPOT FILE
406  002040' 012701  000444'                 MOV     #RBUF,R1        ;INPUT BUFFER
407  002044' 010102                          MOV     R1,R2           ;END ADDRESS
408  002046' 062702  001000                  ADD     #512.,R2        ;LAST LOCATION OF THE INPUT BUFFER
409  002052' 062701  000010                  ADD     #10,R1          ;BAD SOPT FILE STARTS FROM 5TH WORDS
410  002056' 012703  000252'                 MOV     #BADSPT,R3      ;BADSPT RELOCATABLE,TABLE ADDRESS
411  002062' 010304                          MOV     R3,R4           ;LAST LOCATION OF THE TABLE
412  002064' 062704  000040                  ADD     #32.,R4         ;16 WORDS,
413  002070' 022711  000632          1$:     CMP     #410.,(R1)      ;BLOCK OVER CYL 410,THEN DON'T WORRY
414  002074' 101433                          BLOS    6$              ;INCREMENT THE POINTER
415  002076' 005067  176274                  CLR     BLK             ;INITIATE BLOCK COUNTER
416  002102' 011100                          MOV     (R1),R0         ;CYLINDER ADDRESS
417  002104' 001405                          BEQ     3$              ;BRANCH , IF CYL 0
418  002106' 062767  000240  176262  2$:     ADD     #32.*5,BLK      ;BLOCK # FOR ONE CYLINDER
419  002114' 005300                          DEC     R0              ;DECREMENT TEMP COUNT
420  002116' 001373                          BNE     2$              ;LOOP IF NOT ZERO
421  002120' 116100  000003          3$:     MOVB    3(R1),R0        ;TRACK NUMBER
422  002124' 001405                          BEQ     5$              ;BRANCH IF TRACK 0
423  002126' 062767  000040  176242  4$:     ADD     #32.,BLK        ;ADJUST BLOCK FOR ONE TRACK
424  002134' 005300                          DEC     R0              ;BRANCH IF NOT DONE
425  002136' 001373                          BNE     4$              ;
```

```
  426  002140' 116100  000002          5$:     MOVB    2(R1),R0        ;SECTOR NUMBER
  427  002144' 060067  176226                  ADD     R0,BLK          ;UPDATE SECTOR # FOR BLOCK COUNT
  428  002150' 005713                          TST     (R3)            ;ENTRY FOR THE TABLE ?
  429  002152' 001016                          BNE     8$              ;BRANCH IF NOT
  430  002154' 016713  176216                  MOV     BLK,(R3)        ;LOAD INFO TABLE
  431  002160' 062703  000002                  ADD     #2,R3           ;UPDATE TABLE POINTER
  432  002164' 062701  000004          6$:     ADD     #4,R1           ;UPDATE THE BAD SPOT FILE POINTER
  433  002170' 020102                  7$:     CMP     R1,R2           ;END OF FILE
  434  002172' 103011                          BHIS    9$              ;BRANCH IF IT IS
  435  002174' 020304                          CMP     R3,R4           ;END OF TABLE ?
  436  002176' 103734                          BLO     1$              ;BRANCH IF NOT
  437  002200' 104403  006766' 000000'         MSGN$,MES15,BEGIN
  438  002206' 000403                          BR      9$      ;DON'T DON ANYTHING
  439  002210' 062703  000002          8$:     ADD     #2,R3   ;UPDTAE THE TABLE ENTRY
  440  002214' 000765                          BR      7$      ;CHECK THE TABLE AND FILE ENTRY
  441  002216' 000655                  9$:     BR      RES2X   ;TO NEXT DRIVE
  442  002220' 012767  000001  176150  10$:    MOV     #1,BLK  ;RESET BLOCK NUMBER
  443  002226' 012767  000001  176152          MOV     #1,FLAG         ;RESTORE FIRST TIME FLAG
  444                                                          ;COUNTINUE
  445  002234' 000410                          BR       RSTRT1         ;+ SUPPORT - DT03
  446
  447  002236' 005767  176156          RESTRT: TST     CNT1            ;+      / SUPPORT
  448  002242' 001005                          BNE     RSTRT1          ;+     /  FOR
  449  002244' 005767  176146                  TST     CNT             ;+    /   DT03
  450  002250' 001002                          BNE     RSTRT1          ;+   /    BUS
  451  002252' 000167  177326                  JMP     START           ;+  /     SWITCH
  452  002256'                 RSTRT1:                                 ;+/
  453  002256' 104415  000000' 000124'         GETPA$,BEGIN, RBUFVA            ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
  454  002264' 005067  176126                  CLR     CNT             ;CLEAR THE LOCATE COUNTER
  455  002270' 012767  177777  176116  LOOP1:  MOV     #-1,UNITNO      ;PRE-SET UNIT NUMBER
  456  002276' 104414  000000'                 GWBUF$, BEGIN           ;GET WRITE BUFFER INFORMATION
  457  002302' 004767  002244                  JSR     PC,PICKBK       ;GO PICK A BLK
  458  002306'                 LOOP2:
  459  002306' 004767  002122          3$:     JSR     PC,PICKDR       ;GO PICK A DRIVE
  460  002312' 000405                          BR      1$              ;RETURNS HERE IF ALL DRIVES DONE
  461  002314' 005067  176116                  CLR     TRY             ;ELSE CLR RE TRY COUNT
  462  002320' 004767  000034                  JSR     PC,CYCLE        ;GO DO A CYCLE ON THIS DRIVE
  463  002324' 000770                          BR      LOOP2           ;DO IT TO NEXT DRIVE
  464
  465  002326' 042767  000001  176052  1$:     BIC     #1,FLAG         ;CLEAR FIRST TIME FLAG
  466  002334' 005767  176050                  TST     DVICE           ;ANYBODY LEFT TO CHECK?
  467  002340' 001002                          BNE     2$              ;BR IF YES
  468  002342' 104410  000000'                 END$,BEGIN              ;
  469  002346' 005267  176046          2$:     INC     CNT1            ;+BUMP COUNTER FOR DT03
  470  002352' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
  471                                                                  ;MONITOR SHALL TEST END OF PASS
  472  002356' 000744                          BR      LOOP1
```



```
  473
  474  002360' 032767  000020  175430  CYCLE:  BIT     #20,SR1         ;B-PORT?
  475  002366' 001114                          BNE     CYCLE8          ;BR IF SO, ELSE DO A-PORT
  476  002370' 004567  003064                  JSR     R5,READY        ;READY ?
  477  002374' 000402                          BR      CYA             ;YES
  478  002376' 004767  002620                  JSR     PC,NOTRDY       ;NO
  479  002402' 052767  000004  175776  CYA:    BIS     #4,FLAG         ;SET THE A-PORT FLAG
  480  002410' 004767  001636                  JSR     PC,BLKADR       ;CONVERT BLK TO DISK ADDR
  481  002414' 004567  000452                  JSR     R5,WRITE        ;GO WRITE A BLOCK
  482  002420' 005767  176016                  TST     BADSEC          ;FOUND BAD SPOT ?
  483  002424' 001401                          BEQ     .+4             ;BRANCH IF NOT
  484  002426' 000207                          RTS     PC              ;OTHERWISE EXIT
  485  002430' 004567  000532                  JSR     R5,WRIICK       ;GO DO WRITE CHECK
  486  002434' 004567  000622                  JSR     R5,READ         ;GO READ A BLOCK
  487  002440' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
  488  002446' 002450'                         .+2                     ; IF ERROR, CONTINUE
  489
  490                                  ;NOW UPDATE BLOCK 0 FOR B-PORT'S INFORMATION
  491
  492  002450' 004767  002154                  JSR     PC,CLRRB        ;CLEAR THE READ BUFFER
  493  002454' 016767  175726  175762          MOV     FLAG,RBUF       ;PUT INFO IN FIRST WORD
  494  002462' 016767  175710  175756          MOV     BLK,RBUF+2      ;PUT CURRENT ADDR IN NXT
  495  002470' 004567  001052                  JSR     R5,WRT00        ;GO UPDATE BLOCK 0
  496  002474' 012767  000000  175724          MOV     #0,TIMER        ;INIT WAIT LOOP
  497  002502' 012777  000013  177010  1$:     MOV     #13,@RMCS1      ;GIVE THE DRIVE TO B
  498  002510' 005367  175712          2$:     DEC     TIMER
  499  002514' 001006                          BNE     3$
  500  002516' 004767  002126                  JSR     PC,DROP
  501  002522' 104403  000000' 006744'         MSGN$,BEGIN,PORTHG      ;ASCII MESSAGE CALL WITH COMMON HEADER
  502  002530' 000207                          RTS     PC
  503  002532' 012700  000002          3$:     MOV     #2,R0
  504  002536'                          4$:
  505  002536' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
  506  002542' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
  507  002546' 005300                          DEC     R0
  508  002550' 001372                          BNE     4$
  509  002552' 004567  002702                  JSR     R5,READY        ;READY ?
  510  002556' 000402                          BR      5$              ;YES
  511  002560' 004767  002436                  JSR     PC,NOTRDY       ;NO
  512  002564' 004567  001052          5$:     JSR     R5,RD00         ;GO READ BLOCK 0
  513  002570' 000240                          NOP                     ;CHANGE TO BR IF ONLY SINGLE PORT
  514  002572' 032767  000002  175644          BIT     #2,RBUF         ;HAS B UPDATED BLK 0 YET?
  515  002600' 001740                          BEQ     1$              ;BR BACK IF NOT
  516  002602' 004567  000454                  JSR     R5,READ         ;GO READ THE DATA B WROTE
  517  002606' 104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
  518  002614' 002616'                         .+2                     ; IF ERROR, CONTINUE
  519  002616' 000207                          RTS     PC
```

```
                                ;
002620' 012767  000000  175600  CYCLEB: MOV     #0,TIMER
002626' 004567  002626                  JSR     R5,READY        ;READY ?
002632' 000402                          BR      CYB             ;YES
002634' 004767  002362                  JSR     PC,NOTRDY       ;NO
002640' 004567  000776          CYB:    JSR     R5,RD00         ;GO SEE IF A HAS DONE YET
002644' 032767  000001  175534          BIT     #1,FLAG         ;IS THIS THE FIRST TIME SINCE START?
002652' 001410                          BEQ     1$              ;BR IF NO, SKIP THIS CHECK
002654' 032767  000004  175562          BIT     #4,RBUF         ;HAS A WRITTEN THIS BLOCK?
002662' 001410                          BEQ     2$              ;BR IF NO
002664' 032767  000001  175552          BIT     #1,RBUF         ;HAS A SET THE FIRST TIME FLAG?
002672' 001404                          BEQ     2$              ;BR IF NOT, MUST WAIT
002674' 032767  000002  175542  1$:     BIT     #2,RBUF         ;HAS A WRITTEN THIS BLOCK SINCE B DID?
002702' 001432                          BEQ     3$              ;BR IF SO, ELSE
002704' 012777  000013  176606  2$:     MOV     #13,@RMCS1      ;GIVE PORT TO A
002712' 012700  000002                  MOV     #2,R0
002716'                         4$:
002716' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002722' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002726' 005300                          DEC     R0
002730' 001372                          BNE     4$
002732' 005367  175470                  DEC     TIMER
002736' 001006                          BNE     6$
002740' 004767  001704                  JSR     PC,DROP
002744' 104403  000000' 006744'         MSG$,BEGIN,PORTHG       ;ASCII MESSAGE CALL WITH COMMON HEADER
002752' 000207                          RTS     PC
002754' 004567  002500          6$:     JSR     R5,READY        ;READY ?
002760' 000727                          BR      CYB             ;YES
002762' 004767  002234                  JSR     PC,NOTRDY       ;NO
002766' 000724                          BR      CYB             ;TRY AGAIN
                                ;
002770' 016767  175452  175400  3$:     MOV     RBUF+2,BLK      ;GET THE CURRENT BLK
002776' 004767  001250                  JSR     PC,BLKADR       ;GENERATE DISK ADDR FROM IT
003002' 004567  000254                  JSR     R5,READ         ;GO READ WHAT A WROTE
003006' 004567  000344                  JSR     R5,WRITEB       ;GO WRITE IT BACK OUT
003012' 004567  000434                  JSR     R5,WRITCB       ;GO WRITE CHECK IT
003016' 004767  001606                  JSR     PC,CLRRB        ;CLEAR BUFFER
003022' 052767  000002  175356          BIS     #2,FLAG         ;SET BIT SAYING B'S DONE
003030' 016767  175352  175406          MOV     FLAG,RBUF       ;PUT INFO INTO RBUF
003036' 004567  000504                  JSR     R5,WRT00        ;GO WRITE IT FOR A TO SEE
003042' 012777  000013  176450          MOV     #13,@RMCS1      ;GIVE DRIVE TO A
003050' 012700  000002                  MOV     #2,R0
003054'                         5$:
003054' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003060' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003064' 005300                          DEC     R0
003066' 001372                          BNE     5$
003070' 000207                          RTS     PC
```



```
                                        .MACRO  LINEUP  EABITS   ; LINE UP EA BITS FOR RHCS1
                                ;       LINEUP  EABITS          ; LINE UP EA BITS FOR RHCS1
                                        MOV     EABITS,R0       ; GET EXTENDED MEMORY BITS
                                        ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
                                        ASL     R0              ; TO LINE UP WITH RHCS1
                                        ASL     R0              ;
                                        ASL     R0              ;
                                        MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
                                        .ENDM   LINEUP
                                ;
                                ;
003072' 012767  000161  175324  WRITE:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
003100' 012767  003072' 175324          MOV     #WRITE,FERADR   ;SAVE WHERE WE WERE
003106' 016746  175030                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
003112' 005416                          NEG     (SP)            ;NEGATE IT
003114' 012677  176402                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
003120' 016777  175010  176376          MOV     WBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003126' 016777  175246  176372          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003134' 016777  175242  176412          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
003142' 016700  174770                  MOV     WBUFEA,R0       ; GET EXTENDED MEMORY BITS
003146' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003150' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003152' 006300                          ASL     R0              ;
003154' 006300                          ASL     R0              ;
003156' 010067  175206                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003162' 000167  000556                  JMP     GO              ; CONTINUE
003166' 012767  000151  175230  WRITCK: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
003174' 012767  003166' 175230          MOV     #WRITCK,FERADR  ;SAVE WHERE WE WERE
003202' 016746  174734                  MOV     WBUFSZ,-(SP)    ;GET WRITE SIZE
003206' 005416                          NEG     (SP)            ;NEGATE IT
003210' 012677  176306                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
003214' 016777  174714  176302          MOV     WBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003222' 016777  175152  176276          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003230' 016777  175146  176316          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
                                ;       LINEUP  WBUFEA          ; LINE UP EA BITS FOR RHCS1
003236' 016700  174674                  MOV     WBUFEA,R0       ; GET EXTENDED MEMORY BITS
003242' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003244' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003246' 006300                          ASL     R0              ;
003250' 006300                          ASL     R0              ;
003252' 010067  175112                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
003256' 000167  000462                  JMP     GO              ; CONTINUE
003262' 012767  000171  175134  READ:   MOV     #171,FUNC       ; LOAD READ FUNCTION
003270' 012767  003262' 175134          MOV     #READ,FERADR    ;SAVE WHERE WE WERE
003276' 016746  174630                  MOV     RBUFSZ,-(SP)    ;GET READ SIZE
003302' 005416                          NEG     (SP)            ;NEGATE IT
003304' 012677  176212                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
003310' 016777  174612  176206          MOV     RBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
003316' 016777  175056  176202          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
003324' 016777  175052  176222          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
                                ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
003332' 016700  174572                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
003336' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
003340' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
003342' 006300                          ASL     R0              ;
```

```
624  003344' 006300                          ASL     R0              ;
625  003346' 010067  175016                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
626  003352' 000167  000366                  JMP     GO              ; CONTINUE
627  003356' 012767  000161  175040  WRITEB: MOV     #161,FUNC       ; LOAD WRITE FUNCTION
628  003364' 012767  003356' 175040          MOV     #WRITEB,FERADR  ;SAVE WHERE WE WERE
629  003372' 016746  174534                  MOV     RBUFSZ,-(SP)    ;GET WRITE SIZE
630  003376' 005416                          NEG     (SP)            ;NEGATE IT
631  003400' 012677  176116                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
632  003404' 016777  174516  176112          MOV     RBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
633  003412' 016777  174762  176106          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
634  003420' 016777  174756  176126          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
635                          ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
636  003426' 016700  174476                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
637  003432' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
638  003434' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
639  003436' 006300                          ASL     R0              ;
640  003440' 006300                          ASL     R0              ;
641  003442' 010067  174722                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
642  003446' 000167  000272                  JMP     GO              ; CONTINUE
643  003452' 012767  000151  174744  WRITCB: MOV     #151,FUNC       ; LOAD WRITE-CHECK FUNCTION
644  003460' 012767  003452' 174744          MOV     #WRITCB,FERADR  ;SAVE WHERE WE WERE
645  003466' 016746  174440                  MOV     RBUFSZ,-(SP)    ;GET WRITE SIZE
646  003472' 005416                          NEG     (SP)            ;NEGATE IT
647  003474' 012677  176022                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
648  003500' 016777  174422  176016          MOV     RBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
649  003506' 016777  174466  176012          MOV     DSKADR,@RMDA    ; LOAD DISK ADDRESS
650  003514' 016777  174662  176032          MOV     CYL,@RMDC       ; LOAD CYLINDER ADDRESS
651                          ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
652  003522' 016700  174402                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
653  003526' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
654  003530' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
655  003532' 006300                          ASL     R0              ;
656  003534' 006300                          ASL     R0              ;
657  003536' 010067  174626                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
658  003542' 000167  000176                  JMP     GO              ; CONTINUE
659  003546' 012767  000161  174650  WRT00:  MOV     #161,FUNC       ; LOAD WRITE FUNCTION
660  003554' 012767  003546' 174650          MOV     #WRT00,FERADR   ;SAVE WHERE WE WERE
661  003562' 016746  174344          WRT00R: MOV     RBUFSZ,-(SP)    ;GET WRITE SIZE
662  003566' 005416                          NEG     (SP)            ;NEGATE IT
663  003570' 012677  175726                  MOV     (SP)+,@RMWC     ; LOAD WORD COUNT
664  003574' 016777  174326  175722          MOV     RBUFPA,@RMBA    ; LOAD BUFFER ADDRESS
665  003602' 012777  000000  175716          MOV     #0,@RMDA        ; LOAD DISK ADDRESS
666  003610' 012777  000000  175736          MOV     #0,@RMDC        ; LOAD CYLINDER ADDRESS
667                          ;       LINEUP  RBUFEA          ; LINE UP EA BITS FOR RHCS1
668  003616' 016700  174306                  MOV     RBUFEA,R0       ; GET EXTENDED MEMORY BITS
669  003622' 006300                          ASL     R0              ; SHIFT 4 PLACES TO THE LEFT
670  003624' 006300                          ASL     R0              ; TO LINE UP WITH RHCS1
671  003626' 006300                          ASL     R0              ;
672  003630' 006300                          ASL     R0              ;
673  003632' 010067  174532                  MOV     R0,XMEM         ; SAVE THE SHIFTED BITS
674  003636' 000167  000102                  JMP     GO              ; CONTINUE
675  003642' 012767  000171  174554  RD00:   MOV     #171,FUNC       ;SET FUNCTION TO A READ
676  003650' 012767  003642' 174554          MOV     #RD00,FERADR    ;SAVE WHERE WE WERE
677  003656' 000741                          BR      WRT00R          ;THEN USE WRT00 ROUTINE FOR THE READ
678
679  003660' 016777  174530  175642  CLEAR:  MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
```



```
680  003666' 012777  000011  175624          MOV     #11,@RMCS1      ; ISSUE A DRIVE CLEAR
681  003674' 000240                          NOP                     ;WAIT
682  003676' 000240                          NOP                     ;FOR DRIVE CLEAR TO FINISH
683  003700' 012777  000023  175612          MOV     #23,@RMCS1      ;ISSUE A PACK ACK
684  003706' 105777  175606          1$:     TSTB    @RMCS1
685  003712' 100401                          BMI     2$
686  003714' 000774                          BR      1$              ; NO, WAIT TILL DONE
687  003716' 017746  175614          2$:     MOV     @RMAS,-(SP)     ;CLEAR AS BIT
688  003722' 012677  175610                  MOV     (SP)+,@RMAS
689  003726' 012777  040000  175564          MOV     #BIT14,@RMCS1   ; CLEAR ANY CONTROLLER ERRORS
690  003734' 012777  010000  175610          MOV     #BIT12,@RMOF    ; SET BIT FOR 11 FORMAT
691  003742' 000205                          RTS     R5              ; RETURN
692
693  003744' 016777  174444  175556  GO:     MOV     UNITNO,@RMCS2   ; LOAD UNIT SELECT
694  003752' 032767  001000  174076          BIT     #ADDR22,RES1    ;11/70 MONITOR?
695  003760' 001434                          BEQ     1$              ;NO
696  003762' 017767  175536  174376          MOV     @RMBA,PA18      ;GET 18 BIT ADDR
697  003770' 006267  174374                  ASR     XMEM            ;SHIFT EA BITS TO POSITION 4,5
698  003774' 006267  174370                  ASR     XMEM
699  004000' 006267  174364                  ASR     XMEM
700  004004' 006267  174360                  ASR     XMEM
701  004010' 104416  000000' 000366'        MAP22$, BEGIN,PA18              ; GET 22-BIT ADDR FROM 18-BIT ADDR
702  004016' 016777  174350  175500          MOV     PA22,@RMBA      ;LOAD BA REG
703  004024' 016777  174344  175536          MOV     EA22,@RMBAE     ;LOAD BAE REG
704  004032' 042767  000034  174334          BIC     #34,EA22        ;CLEAR UNWANTED BITS
705  004040' 000367  174330                  SWAB    EA22            ;LOAD INTO BITS 8,9
706  004044' 016767  174324  174316          MOV     EA22,XMEM       ;LOAD XMEM TO SET INTO FUNCTION CODE
707  004052' 056767  174312  174344  1$:     BIS     XMEM,FUNC       ; LOAD EXTENDED MEMORY BITS
708  004060' 016777  174340  175432          MOV     FUNC,@RMCS1     ; EXECUTE THE FUNCTION
709  004066' 104400  000000'                 EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
710  004072' 010046                  INTRUPT: MOV    R0,-(SP)        ;SAVE R0
711  004074' 016700  174314                  MOV     UNITNO,R0       ;DRIVE #
712  004100' 116000  004542'                 MOVB    BITTAB(R0),R0   ;DRIVE INDICATOR
713  004104' 017746  175426                  MOV     @RMAS,-(SP)     ;CLEAR CORESPONDING
714  004110' 040016                          BIC     R0,(SP)         ;ATTENTION BIT
715  004112' 012677  175420                  MOV     (R6)+,@RMAS     ;IN THE ATTENTION SUMMARY REGISTER
716  004116' 012600                          MOV     (SP)+,R0        ;RESTORE R0
717                                          ;-----------------------------------------------------------------
718  004120' 000004  000000' 004126'         PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
719                                          ;-----------------------------------------------------------------
720
721  004126' 004567  000554          1$:     JSR     R5,ERRORS       ; GO CHECK FOR ERRORS
722  004132' 000401                          BR      2$              ;ERRORS DETECTED
723  004134' 000205                          RTS     R5              ;OTHERWISE,  RETURN OK
724
725  004136' 005267  174274          2$:     INC     TRY             ;COUNT AN ERROR
726  004142' 026727  174270  000004          CMP     TRY,#4          ;TOO MANY FOR THIS CYCLE?
727  004150' 001035                          BNE     4$              ;BR IF NOT
728  004152' 104403  000000' 006724'         MSG$,BEGIN,EXCED        ;ASCII MESSAGE CALL WITH COMMON HEADER
729  004160' 012605                          MOV     (SP)+,R5        ;FOR RETURN WE WONT DO
730  004162' 032767  000020  173626          BIT     #20,SR1
731  004170' 001411                          BEQ     3$
732  004172' 004767  000452                  JSR     PC,DROP
733  004176' 104403  000000' 006756'         MSG$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
734  004204' 016706  173622                  MOV     SPOINT,R6
735  004210' 000167  176072                  JMP     LOOP2
```

```
                                ;IN PROCESS OF RETRIVING BAD SPOT ?
004214' 022767 000770 174164 3$:    CMP     #770,FLAG
004222' 001002                      BNE     .+6             ;BRANCH IF NOT
004224' 000167 175522               JMP     RES2X           ;IF SO, THEN BRANCH BACK
004230' 004767 000316               JSR     PC,PICKBK       ;TRY A DIFFERENT BLOCK
004234' 005367 174154               DEC     UNITNO          ;WANT TO RE-DO SAME DRIVE WERE ON
004240' 000167 176042               JMP     LOOP2           ;GO DO IT
004244' 162705 000004       4$:     SUB     #4,R5           ;BUMP BACK RETURN FOR A RETRY
004250' 000205                      RTS     R5

                            ;       ------------------------

004252' 005067 174126       BLKADR: CLR     TRACK           ;START ADDR AT ZERO
004256' 005067 174116               CLR     DSKADR
004262' 005067 174114               CLR     CYL
004266' 005767 174146               TST     MIXDV           ;MIXED DRIVE ?
004272' 100430                      BMI     5$              ;BRANCH IF SO
004274' 016700 174076               MOV     BLK,R0          ;GET BLOCK NUMBER
004300' 162700 000240       1$:     SUB     #32.*5.,R0              ;SUBTRACT A CYLINDER'S WORTH OF SECTORS
004304' 100403                      BMI     2$              ;BR OUT IF WENT TOO FAR
004306' 005267 174070               INC     CYL             ;ELSE COUNT A CYLINDER
004312' 000772                      BR      1$              ;AND DO IT AGAIN
004314' 062700 000240       2$:     ADD     #32.*5.,R0              ;PUT BACK ONE
004320' 162700 000040       3$:     SUB     #32.,R0         ;SUBTRACT A TRACK'S WORTH OF SECTORS
004324' 100403                      BMI     4$              ;BRANCH IF WE WENT TO FAR
004326' 105267 174053               INCB    TRACK+1         ;ELSE COUNT ANOTHER TRACK
004332' 000772                      BR      3$              ;AND DO IT AGAIN
004334' 062700 000040       4$:     ADD     #32.,R0         ;PUT BACK ONE
004340' 010067 174034               MOV     R0,DSKADR       ;PUT IN SECTOR ADDRESS
004344' 116767 174035 174027        MOVB    TRACK+1,DSKADR+1        ;PUT IN TRACK ADDR
004352' 000427                      BR      10$             ;EXIT
004354' 016700 174016       5$:     MOV     BLK,R0          ;LOAD THE BLOCK NUMBER
004360' 162700 000642       6$:     SUB     #22.*19.,R0     ;22 SECTOR*19 TRACK
004364' 100403                      BMI     7$              ;DONE IF NEG
004366' 005267 174010               INC     CYL             ;CYLINDER ADDRESS
004372' 000772                      BR      6$              ;LOOPING BACK
004374' 062700 000642       7$:     ADD     #22.*19.,R0     ;ADJUST ONE CYLINDER
004400' 162700 000026       8$:     SUB     #22.,R0         ;FIND WHICH TRACK
004404' 100403                      BMI     9$              ;EXIT IF DONE
004406' 105267 173773               INCB    TRACK+1         ;INCREMENT TRACK COUNT
004412' 000772                      BR      8$              ;BRANCH IF NOT DONE
004414' 062700 000026       9$:     ADD     #22.,R0         ;ADJUST ONE TRACK
004420' 010067 173754               MOV     R0,DSKADR       ;TRACK AND SECTOR ADDRESSES
004424' 116767 173755 173747        MOVB    TRACK+1,DSKADR+1        ;
004432' 000207              10$:    RTS     PC

004434'                     PICKDR:
004434' 005067 174000               CLR     MIXDV           ;RESET  THE MIX DRIVE FLAG
004440' 005267 173750       1$:     INC     UNITNO          ;POINT TO NEXT DRIVE
004444' 026727 173744 000010        CMP     UNITNO,#8.              ;DONE LOOKING?
004452' 002401                      BLT     2$              ;BR IF NO, ELSE
004454' 000207                      RTS     PC
004456' 016700 173732       2$:     MOV     UNITNO,R0       ;USE AS AN INDEX
004462' 136067 004542' 173720       BITB    BITTAB(R0),DVICE        ;TEST THIS DEVICE?
004470' 001001                      BNE     3$              ;BR IF YES, ELSE
004472' 000762                      BR      1$
004474' 062716 000002       3$:     ADD     #2,(SP)         ;BUMP FOR A GOOD RETURN
```



```
004500' 004567 000754               JSR     R5,READY        ;SEE IF DRIVE IS READY
004504' 000402                      BR      4$              ;BR IF IT WAS READY
004506' 004767 000510               JSR     PC,NOTRDY       ;ELSE GO CLEAR IT AND CHECK AGAIN
004512' 022777 024025 175026 4$:    CMP     #24025,@RMDT    ;RM02 ?
004520' 001407                      BEQ     5$              ;EXIT IF SO
004522' 022777 024024 175016        CMP     #24024,@RMDT    ;RM03 ?
004530' 001403                      BEQ     5$              ;BRANCH IF SO
004532' 012767 177777 173700        MOV     #-1,MIXDV       ;OTHERWISE SET THE FLAG
004540' 000207              5$:     RTS     PC              ;RETURN WITH A UNITNO READY
004542' 001001              BITTAB: 1001
004544' 004004                      4004
004546' 020020                      20020
004550' 100100                      100100

004552' 062767 000001 173616 PICKBK: ADD    #1,BLK          ;DO NEXT BLOCK(SECTOR)
004560' 026727 173612 077777        CMP     BLK,#77777
004566' 103403                      BLO     1$
004570' 012767 000001 173600        MOV     #1,BLK          ;GO BACK TO BLOCK 1
004576' 005767 173636       1$:     TST     MIXDV           ;NOT RM03/RM02 ?
004602' 100411                      BMI     3$              ;BRANCH IF NOT
004604' 012700 000252'              MOV     #BADSPT,R0      ;GET BAD SPOT TABLE
004610' 012701 000020               MOV     #16.,R1         ;LOOK FOR 16 ENTRIES
004614' 026720 173556       2$:     CMP     BLK,(R0)+       ;IS THIS A BAD BLK?
004620' 001754                      BEQ     PICKBK          ;IF YES, GO PICK A NEW ONE
004622' 005301                      DEC     R1              ;COUNT A TABLE LOOK-UP
004624' 001373                      BNE     2$              ;BR BACK IF MORE TO GO
004626' 000207              3$:     RTS     PC

004630' 012700 000444'      CLRRB:  MOV     #RBUF,R0        ;CLEAR RBUF BUFFER
004634' 016701 173272               MOV     RBUFSZ,R1       ;GET ITS ADDR AND SIZE
004640' 005020              CLRCOM: CLR     (R0)+           ;CLEAR ANOTHER
004642' 005301                      DEC     R1              ;COUNT ANOTHER
004644' 001375                      BNE     CLRCOM          ;BR BACK TILL DONE
004646' 000207                      RTS     PC

004650' 012701 000001       DROP:   MOV     #1,R1           ; INITIALIZE DROP PICKER
004654' 016700 173534               MOV     UNITNO,R0       ; GET THE DRIVE NUMBER
004660' 001403                      BEQ     2$              ; IF DRIVE 0 GO DROP IT
004662' 006301              1$:     ASL     R1              ; POINT TO NEXT DRIVE
004664' 005300                      DEC     R0              ; IS THIS THE ONE ?
004666' 001375                      BNE     1$              ; NO, LOOK AGAIN
004670' 040167 173514       2$:     BIC     R1,DVICE        ; DROP THE DRIVE
                                    ;*****************************************************************
                                                            ;CONVERT UNITNO TO ASCII AND
                                                            ;STORE AT ADR1
004674' 104420 000000' 000414'      OTOAS,BEGIN,UNITNO,ADR1
004702' 006772'
                                    ;*****************************************************************
004704' 000207                      RTS     PC              ; RETURN
                            ;       ------------------------

004706' 005067 173530       ERRORS: CLR     BADSEC          ;
004712' 005777 174602               TST     @RMCS1          ; ATTENTION OR ERROR ?
004716' 100402                      BMI     1$              ;BR IF YES
004720' 005725                      TST     (R5)+           ;BUMP FOR GOOD RETURN
```

```
                                        RTS     R5              ;AND RETURN
                                1$:     CLR     R0
004726' 016701  174566                  MOV     RMCS1,R1        ;ADDRESS OF RMCS1
004732' 012160  000312'         22$:    MOV     (R1)+,S(R0)     ;READ AND STORE
004736' 005720                          TST     (R0)+           ;ALL CONTROLLER AND RM03
004740' 022700  000046                  CMP     #46,R0          ;REGISTERS
004744' 001372                          BNE     22$
004746' 004767  000660                  JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
004752' 005767  173344                  TST     S+10            ;IS THIS A DATA LATE ERROR?
004756' 100012                          BPL     11$             ;NO
004760' 005267  173436                  INC     DLTCNT          ;ADD 1 TO DATA LATE COUNTER
004764' 032767  000004  173024          BIT     #BIT2,SR1       ;TYPE ERROR AND COUNT IT?
004772' 001103                          BNE     8$              ;NO
004774' 104403  000000' 006730'         MSGN$,BEGIN,DLTERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
005002' 000450                          BR      5$              ;CONT
005004' 032767  020000  173300  11$:    BIT     #BIT13,S        ; MASSBUS CONTROL PARITY ERROR ?
005012' 001035                          BNE     3$              ; YES
005014' 032767  000400  173300          BIT     #BIT8,S+10      ; MASSBUS DATA PARITY ERROR ?
005022' 001035                          BNE     4$              ; YES
005024' 032767  040000  173260          BIT     #BIT14,S        ; TRANSFER ERROR ?
005032' 001015                          BNE     2$              ; YES
005034' 032767  040000  173262          BIT     #BIT14,S+12     ; ANY DRIVE ERRORS ?
005042' 001030                          BNE     5$              ; YES
005044' 005767  173260                  TST     S+16            ; ANY ATTENTIONS ACTIVE ?
005050' 001025                          BNE     5$              ; YES, CONTINUE
005052' 005067  173030                  CLR     ERRTYP          ;UNKNOWN ERROR
                                        ;****************************************************************
005056' 104405  000000' 001444'         HRDER$,BEGIN,TABLE      ; SPECIAL CONDITION SET BUT NO REASON FOUND
                                        ;****************************************************************
005064' 000446                          BR      8$              ; RETURN
005066'                         2$:
005066' 032767  100000  173260          BIT     #BIT15,S+42     ;IS A BAD SPOT ******* ?
005074' 001045                          BNE     9$              ;BRANCH IF SO *****
005076' 104403  000000' 006710'         MSGN$,BEGIN,TRERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
005104' 000407                          BR      5$              ; GO DUMP REGISTERS
005106'                         3$:
005106' 104403  000000' 006714'         MSGN$,BEGIN,MCPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
005114' 000403                          BR      5$              ; GO DUMP REGISTERS
005116'                         4$:
005116' 104403  000000' 006720'         MSGN$,BEGIN,MDPERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
005124' 005767  173200          5$:     TST     S+16            ; ANY ATTENTIONS ACTIVE ?
005130' 001004                          BNE     6$              ; YES, CONTINUE
005132' 016777  173172  174376          MOV     S+16,@RMAS      ;KILL ATT BITS
005140' 000205                          RTS     R5
005142' 016700  174374          6$:     MOV     RMDB,R0         ; SAVE ADDRESS OF DATA BUFFER
005146' 105767  173150                  TSTB    S+10            ; CAN DATA BUFFER BE READ ?
005152' 100403                          BMI     7$              ; YES, CONTINUE
005154' 012767  000430' 174360          MOV     #ZERO,RMDB      ; NO, LOAD ADDRESS OF ZERO
005162' 012767  000001  172716  7$:     MOV     #1,ERRTYP       ;DATA ERROR
                                        ;****************************************************************
005170' 104405  000000' 001444'         HRDER$,BEGIN,TABLE      ; DUMP RH11 AND RP REGISTERS
                                        ;****************************************************************
005176' 010067  174340                  MOV     R0,RMDB         ; RESTORE DATA BUFFER ADDRESS
005202' 004767  000014          8$:     JSR     PC,NOTRDY       ;GO CLEAR OUT ERRORS
005206' 000205                          RTS     R5              ; ERRORS DETECTED, RETURN
005210' 005725                  9$:     TST     (R5)+           ;IF BAD SOFT DON'T REPORT
```



```
005212' 012767  177777  173222            MOV     #-1,BADSEC      ;SET BAD SECTOR DETECT FLAG
005220' 000205                            RTS     R5      ;EXIT*****


005222' 012767  077777  173204  NOTRDY:   MOV     #77777,CLK      ; SET THE TIMER
005230' 016777  173160  174272  4$:       MOV     UNITNO,@RMCS2   ;MOVE DRIVE NUMBER TO RH11/70
005236' 012777  000011  174254            MOV     #11,@RMCS1      ;ISSUE DIRVE CLEAR COMMAND
005244' 032777  004000  174246            BIT     #BIT11,@RMCS1   ;DO I HAVE THE DRIVE DVA?
005252' 001406                            BEQ     6$              ;NO
005254' 010046                            MOV     R0,-(SP)
005256' 012600                            MOV     (SP)+,R0        ;JUST WASTE A LITTLE TIME
005260' 032777  004000  174232            BIT     #BIT11,@RMCS1           ;STILL GOT IT?
005266' 001023                            BNE     2$
005270'                           6$:
005270' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005274' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005300' 104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
005304' 104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
005310' 032777  010000  174212            BIT     #BIT12,@RMCS2   ;NED SET ?
005316' 001014                            BNE     5$              ;YES
005320' 005367  173110                    DEC     CLK             ;COUNT # OF TRIES
005324' 001341                            BNE     4$              ;NOT DONE YET
005326' 104403  000000' 006734'           MSGN$,BEGIN,NUT ;ASCII MESSAGE CALL WITH COMMON HEADER
005334' 000426                            BR      7$              ;COULD NOT GET DRIVE
005336' 004567  176316          2$:       JSR     R5,CLEAR        ;SET THE CONTROLLER AND DRIVE
005342' 004567  000112                    JSR     R5,READY        ; IS DRIVE READY ?
005346' 000443                            BR      1$              ; YES, CONTINUE
005350' 004767  000256          5$:       JSR     PC,ERSUB1       ; LOAD ERROR INFORMATION
005354' 005000                            CLR     R0              ;MOVE DRIVE REG INTO TABLE
005356' 016701  174136                    MOV     RMCS1,R1
005362' 012160  000312'         22$:      MOV     (R1)+,S(R0)     ;READ AND STORE ALL
005366' 005720                            TST     (R0)+           ;RH11/70 AND RM03
005370' 022700  000046                    CMP     #46,R0          ;REGISTERS
005374' 001372                            BNE     22$
005376' 012767  000006  172502            MOV     #6,ERRTYP       ;DRIVE NOT READY
                                          ;****************************************************************
005404' 104405  000000' 001444'           HRDER$,BEGIN,TABLE      ; DRIVE  NOT READY
                                          ;****************************************************************
005412' 012777  000013  174100  7$:       MOV     #13,@RMCS1      ;RELEASE DRIVE
005420' 004767  177224                    JSR     PC,DROP         ; NO, DROP THE DRIVE
005424' 104403  000000' 006756'           MSGN$,BEGIN,DRP ;ASCII MESSAGE CALL WITH COMMON HEADER
005432' 016706  172374                    MOV     SPOINT,R6
005436' 022767  000770  172742            CMP     #770,FLAG       ;IN PROCESS OF RETRIEVING BAD SPOT
005444' 001002                            BNE     .+6             ;BRANCH IF NOT
005446' 000167  174300                    JMP     RES2X           ;IF SO, BRANCH BACK
005452' 000167  174630                    JMP     LOOP2
005456' 000207                  1$:       RTS     PC              ;RETURN
                                ;         ------------------------


                                ;DVA=0,NED=0     NOT DETERMINE
                                ;DVA=1,NED=0     DRIVE EXIST
                                ;DVA=0,NED=1     DRIVE NOT EXIST
                                ;DVA=1,NED=1     HARDWARE ERROR

005460' 016777  172730  174042  READY:    MOV     UNITNO,@RMCS2   ; LOAD UNIT ADDRESS
```

```
 960  005466' 012777  000011  174024          MOV     #11,@RMCS1      ;CLEAR DRIVE AND GRASP THE DRIVE
 961  005474' 017700  174020                  MOV     @RMCS1,R0       ;R0= BASE ADDRESS OF RH CONTROLLER
 962  005500' 032700  004000                  BIT     #BIT11,R0       ;DVA BIT = 1 ?
 963  005504' 001436                          BEQ     1$              ;NO
 964  00550b' 017700  174020                  MOV     @RMDS,R0        ; SAVE STATUS IN R0
 965  005512' 105700                          TSTB    R0              ; DRIVE READY ?
 966  005514' 100032                          BPL     1$              ; NO
 967  005516' 032700  000100                  BIT     #BIT6,R0        ; VOLUME VALID ?
 968  005522' 001427                          BEQ     1$              ; NO
 969  005524' 032700  000400                  BIT     #BIT8,R0        ; DRIVE PRESENT ?
 970  005530' 001424                          BEQ     1$              ; NO
 971  005532' 032700  004000                  BIT     #BIT11,R0       ; WRITE LOCKED ?
 972  005536' 001021                          BNE     1$              ; YES
 973  005540' 032700  010000                  BIT     #BIT12,R0       ; MEDIUM ON LINE ?
 974  005544' 001416                          BEQ     1$              ; NO
 975  005546' 032700  040000                  BIT     #BIT14,R0       ; ANY ERRORS ?
 976  005552' 001013                          BNE     1$              ; YES
 977  005554' 005700                          TST     R0              ; ATTENTION SET ?
 978  005556' 100411                          BMI     1$              ; YES
 979  005560' 032777  004000  173732          BIT     #BIT11,@RMCS1   ;DVA SET?
 980  005566' 001405                          BEQ     1$              ;BR IF NOT
 981  005570' 032777  010000  173732          BIT     #BIT12,@RMCS2   ;NED BIT SET ?
 982  005576' 001001                          BNE     1$              ;YES,ERROR
 983  005600' 000205                          RTS     R5              ; RETURN READY
 984  005602' 005725                  1$:     TST     (R5)+           ; SKIP INSTRUCTION FOLLOWING CALL
 985  005604' 000205                          RTS     R5              ; RETURN AS NOT READY
 986                                  ;       -------------------------
 987
 988
 989  005606' 014167  172274          ERSUB2: MOV     -(R1),ASB       ; LOAD THE DATA
 990  005612' 010167  172264                  MOV     R1,SBADR        ; LOAD ADDRESS OF DATA WRITTEN
 991  005616' 014267  172266                  MOV     -(R2),AWAS      ; LOAD THE DATA
 992  005622' 010267  172256                  MOV     R2,WASADR       ; LOAD ADDRESS OF DATA READ
 993  005626' 005721                          TST     (R1)+           ; RESET REG. 1
 994  005630' 005722                          TST     (R2)+           ; RESET REG. 2
 995
 996  005632' 016767  173662  172240  ERSUB1: MOV     RMCS1,CSRA      ; LOAD ADR OF CURRENT CSR
 997  005640' 017767  173654  172234          MOV     @RMCS1,ACSR     ; LOAD CONTENTS OF CURRENT CSR
 998  005646' 000207                          RTS     PC              ; RETURN
 999
1000
1001  005650' 017700  173644          REZET:  MOV     @RMCS1,R0       ;DVA BIT OF RMCS1
1002  005654' 032700  004000                  BIT     #BIT11,R0
1003  005660' 001005                          BNE     3$
1004  005662' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1005  005666' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1006  005672' 000766                          BR      REZET
1007  005674' 012777  000040  173626  3$:     MOV     #BIT5,@RMCS2    ; ISSUE AN RM11 INIT
1008  005702' 012767  077777  172524          MOV     #77777,CLK      ; SET THE TIMER
1009  005710' 017700  173604          1$:     MOV     @RMCS1,R0               ; CONTROLLER READY ?
1010  005714' 105700                          TSTB    R0
1011  005716' 100417                          BMI     2$              ; YES, CONTINUE
1012  005720' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
1013  005724' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
1014  005730' 005367  172500                  DEC     CLK             ; WAIT SOME MORE ?
1015  005734' 001365                          BNE     1$              ; YES
```



```
1016  005736' 012767  000004  172142          MOV     #4,ERRTYP       ;CONTROLLER NOT READY
1017                                  ;*****************************************************************
1018  005744' 104405  000000' 000000          HRDERS,BEGIN,NULL       ; CONTROLLER NOT READY
1019                                  ;*****************************************************************
1020  005752' 104410  000000'                 ENDS,BEGIN              ;
1021  005756' 017746  173554          2$:     MOV     @RMAS,-(SP)     ;CLEAR ALL BITS
1022  005762' 012677  173550                  MOV     (SP)+,@RMAS     ;IN THE SUMMARY REGISTER
1023  005766' 000207                          RTS     PC              ; RETURN
1024                                  ;       -------------------------
1025
1026
1027  005770' 016700  172012          SETUP:  MOV     ADDR,R0         ; GET DEVICE ADDRESS
1028  005774' 010067  173520                  MOV     R0,RMCS1        ; GENERATE REGISTER ADDRESSES
1029  006000' 062700  000002                  ADD     #2,R0
1030  006004' 010067  173512                  MOV     R0,RMWC
1031  006010' 062700  000002                  ADD     #2,R0
1032  006014' 010067  173504                  MOV     R0,RMBA
1033  006020' 062700  000002                  ADD     #2,R0
1034  006024' 010067  173476                  MOV     R0,RMDA
1035  006030' 062700  000002                  ADD     #2,R0
1036  006034' 010067  173470                  MOV     R0,RMCS2
1037  006040' 062700  000002                  ADD     #2,R0
1038  006044' 010067  173462                  MOV     R0,RMDS
1039  006050' 062700  000002                  ADD     #2,R0
1040  006054' 010067  173454                  MOV     R0,RMER1
1041  006060' 062700  000002                  ADD     #2,R0
1042  006064' 010067  173446                  MOV     R0,RMAS
1043  006070' 062700  000002                  ADD     #2,R0
1044  006074' 010067  173440                  MOV     R0,RMLA
1045  006100' 062700  000002                  ADD     #2,R0
1046  006104' 010067  173432                  MOV     R0,RMDB
1047  006110' 062700  000002                  ADD     #2,R0
1048  006114' 010067  173424                  MOV     R0,RMMR1
1049  006120' 062700  000002                  ADD     #2,R0
1050  006124' 010067  173416                  MOV     R0,RMDT
1051  006130' 062700  000002                  ADD     #2,R0
1052  006134' 010067  173410                  MOV     R0,RMSN
1053  006140' 062700  000002                  ADD     #2,R0
1054  006144' 010067  173402                  MOV     R0,RMOF
1055  006150' 062700  000002                  ADD     #2,R0
1056  006154' 010067  173374                  MOV     R0,RMDC
1057  006160' 062700  000002                  ADD     #2,R0
1058  006164' 010067  173366                  MOV     R0,RMHR
1059  006170' 062700  000002                  ADD     #2,R0
1060  006174' 010067  173360                  MOV     R0,RMMR2
1061  006200' 062700  000002                  ADD     #2,R0
1062  006204' 010067  173352                  MOV     R0,RMER2
1063  006210' 062700  000002                  ADD     #2,R0
1064  006214' 010067  173344                  MOV     R0,RMEC1
1065  006220' 062700  000002                  ADD     #2,R0
1066  006224' 010067  173336                  MOV     R0,RMEC2
1067  006230' 000207                          RTS     PC
1068  006232' 032767  000001  172146  SETUP2: BIT     #1,FLAG         ;FIRST TIME THROUGH?
1069  006240' 001426                          BEQ     1$              ;BR IF NOT
1070  006242' 016700  171540                  MOV     ADDR,R0         ;BASE ADDRESS OF RH11/RH70
1071  006246' 062700  000046                  ADD     #46,R0          ;INDEX VALUE
```

```
1072  006252' 032767  001000  171576          BIT     #ADDR22,RES1    ;11/70 SUPPORT?
1073  006260' 001416                          BEQ     1$              ;NO
1074  006262' 062700  000002                  ADD     #2,R0
1075  006266' 032767  100000  171522          BIT     #BIT15,SR1      ;SPECIFY 32 REGISTER ON RH70 ?
1076  006274' 001402                          BEQ     3$              ;BRANCH IF NOT
1077  006276' 062700  000024                  ADD     #24,R0          ;OTHERWISE ADJUST THE RMBAE ADDRESS
1078  006302' 010067  173262          3$:     MOV     R0,RMBAE
1079  006306' 062700  000002                  ADD     #2,R0
1080  006312' 010067  173254                  MOV     R0,RMCS3
1081  006316' 016700  171466          1$:     MOV     VECTOR,R0       ; GET VECTOR ADDRESS
1082  006322' 012720  004072'                  MOV     #INTRUPT,(R0)+  ; SET POINTER JUST IN CASE
1083  006326' 116710  171460                  MOVB    BR1,(R0)        ; SET PRIORITY
1084  006332' 000207                  2$:     RTS     PC              ; RETURN
1085
1086  006334' 020040  051124  047101  MES1:   .ASCIZ  '  TRANSFER  ERROR%'
1087  006342' 043123  051105  020040
1088  006350' 051105  047522  022522
1089  006356'     000
1090  006357'     040  046440  051501  MES2:   .ASCIZ  '  MASSBUS  PARITY  ERROR%'
1091  006364' 041123  051525  020040
1092  006372' 040520  044522  054524
1093  006400' 020040  051105  047522
1094  006406' 022522      000
1095  006411'     040  046440  051501  MES3:   .ASCIZ  '  MASSBUS  DATA  PARITY  ERROR%'
1096  006416' 041123  051525  020040
1097  006424' 040504  040524  020040
1098  006432' 040520  044522  054524
1099  006440' 020040  051105  047522
1100  006446' 022522      000
1101  006451'     040  042040  044522  MES4:   .ASCIZ  '  DRIVE  '
1102  006456' 042526  020040      000
1103  006463'     040  042040  047522  MES5:   .ASCIZ  '  DROPPED%'
1104  006470' 050120  042105  000045
1105  006476' 051040  052105  054522  MES6:   .ASCIZ  ' RETRY EXCEEDED%'
1106  006504' 042440  041530  042505
1107  006512' 042504  022504      000
1108  006517'     040  040504  040524  MES10:  .ASCIZ  ' DATA LATE ERROR%'
1109  006524' 046040  052101  020105
1110  006532' 051105  047522  022522
1111  006540'     000
1112  006541'     040  051104  053111  MES11:  .ASCIZ  ' DRIVE NOT READY%'
1113  006546' 020105  047516  020124
1114  006554' 042522  042101  022531
1115  006562'     000
1116  006563'     040  047503  046125  MES12:  .ASCIZ  ' COULD NOT GET DRIVE%'
1117  006570' 020104  047516  020124
1118  006576' 042507  020124  051104
1119  006604' 053111  022505      000
1120  006611'     040  052117  042510  MES14:  .ASCIZ  ' OTHER PORT NOT UPDATING DRIVE%'
1121  006616' 020122  047520  052122
1122  006624' 047040  052117  052440
1123  006632' 042120  052101  047111
1124  006640' 020107  051104  053111
1125  006646' 022505      000
1126  006651'     045  053117  051105  XMES15: .ASCIZ  '%OVER 16 BAD BLOCKS DETECTED%'
1127  006656' 030440  020066  040502
```



```
1128  006664' 020104  046102  041517
1129  006672' 051513  042040  052105
1130  006700' 041505  042524  022504
1131  006706'     000
1132            006710'                         .EVEN
1133  006710' 006334'                  TRERR:  MES1
1134  006712' 177777                          177777
1135  006714' 006357'                  MCPERR: MES2
1136  006716' 177777                          177777
1137  006720' 006411'                  MDPERR: MES3
1138  006722' 177777                          177777
1139  006724' 006476'                  EXCED:  MES6
1140  006726' 177777                          177777
1141  006730' 006517'                  DLTERR: MES10
1142  006732' 177777                          177777
1143
1144  006734' 006541'                  NOT:    MES11
1145  006736' 177777                          177777
1146  006740' 006563'                  TOUT:   MES12
1147  006742' 177777                          177777
1148
1149
1150  006744' 006451'                  PORTHG: MES4
1151  006746' 006777'                          NUMB
1152  006750' 006463'                          MES5
1153  006752' 006611'                          MES14
1154  006754' 177777                          177777
1155  006756' 006451'                  DRP:    MES4
1156  006760' 006777'                          NUMB
1157  006762' 006463'                          MES5
1158  006764' 177777                          177777
1159  006766' 006651'                  MES15:  XMES15
1160  006770' 177777                          177777
1161
1162  006772' 000005                   ADR1:   .BLKB   5
1163  006777'     000                  NUMB:   .BYTE   0
1164  007000'     000                          .BYTE   0
1165            007002'                         .EVEN
1166                                     ;       ------------------------
1167
1168            000001                          .END
```

```
ACSR     000102R          241#    997*
ADDR     000006R          207#   1027    1070
ADDR22=  001000           270#    694    1072
ADR1     006772R          836   1162#
ASB      000106R          245#    989*
ASIAT    000104R          243#
AWAS     000110R          246#    991*
BADSEC   000442R          318#    482     844*    904*
BADSPT   000252R          270#    388     410     813
BEGIN    000000R          204#    383     437     453     456     468     470     487     501     505     506     517     537
                          538     544     563     564     701     709     718     728     733     838     861     875     881
                          884     887     898     918     919     920     921     926     940     944    1004    1005    1012
                         1013    1018    1020
BITTAB   004542R          712     788     801#
BIT0   = 000001           270#
BIT1   = 000002           270#
BIT10  = 002000           270#
BIT11  = 004000           270#    911     915     962     971     979    1002
BIT12  = 010000           270#    690     922     973     981
BIT13  = 020000           270#    863
BIT14  = 040000           270#    669     867     869     975
BIT15  = 100000           270#    879    1075
BIT2   = 000004           270#    859
BIT3   = 000010           270#
BIT4   = 000020           270#
BIT5   = 000040           270#   1007
BIT6   = 000100           270#    967
BIT7   = 000200           270#
BIT8   = 000400           270#    865     969
BIT9   = 001000           270#
BLK      000376R          300#    342     368     375*    415*    418*    423*    427*    430     442*    494     551*    752
                          766     807*    808     810*    815
BLKADR   004252R          480     552     747#
BREAKS=  104407           270#    505     506     537     538     563     564     918     919     920     921    1004    1005
                         1012    1013
BR1      000012R          209#   1083
BR2      000013R          210#
BTOD$  = 104421           270#
CDATA$=  104412           270#    487     517
CDERCT   000144R          260#
CDWDCT   000146R          261#
CLEAR    003660R          679#    928
CLK      000434R          315#    908*    924*   1008*   1014*
CLRCOM   004640R          823#    825
CLRRB    004630R          492     556     821#
CNT      000416R          308#    367     384*    449     454*
CNT1     000420R          309#    447     469*
CONFIG   000056R          229#
CSRA     000100R          239#    996*
CYA      002402R          477     479#
CYB      002640R          523     525*    547     549
CYCLE    002360R          462     474#
CYCLEB   002620R          475     521#
CYL      000402R          302#    397*    586     602     618     634     650     749*    755*    769*
DATCK$=  104411           270#
DATER$=  104404           270#
```



```
DLTCNT   000422R          310#    858*
DLTERR   006730R          861   1141#
DRIVE    000412R          306#
DROP     004650R          500     543     732     828#    943
DRP      006756R          733     944    1155#
DSKADR   000400R          301#    398*    399*    585     601     617     633     649     748*    763*    764*    777*    778*
DVICE    000410R          305#    373*    466     788     834*
DVID1    000414R          211#    373
EA22     000374R          299#    703     704*    705*    706
END1T$=  104413           270#    470
END$   = 104410           270#    468    1020
ERRORS   004706R          721     844#
ERRTYP   000106R          244#    873*    896*    938*   1016*
ERSUB1   005632R          855     931     996#
ERSUB2   005606R          989#
EXCED    006724R          728    1139#
EXIT$  = 104400           270#    709
FERADR   000432R          314#    366     580*    596*    612*    628*    644*    660*    676*
FLAG     000406R          304#    376*    396*    443*    465*    479*    493     526     557*    558     736     946    1068
FREE     000150R          262#
FUNC     000424R          311#    579*    595*    611*    627*    643*    659*    675*    707*    708
GETPA$=  104415           270#    383     453
GO       003744R          594     610     626     642     658     674     693#
GWBUF$=  104414           270#    456
HRDCNT   000044R          224#
HRDER$=  104405           270#    875     898     940    1018
HRDPAS   000050R          226#
ICUNT    000036R          221#
ICOUNT   000040R          222#
IDNUM    000122R          251#
IMODX.=  000000           263#    457
INIT     000030R          218#
INTR     000120R          250#    372*
LOOP1    002270R          386     455#    472
LOOP2    002306R          458#    463     735     741     949
MAP22$=  104416           270#    701
MCPERR   006714R          884   1135#
MDPERR   006720R          887   1137#
MES1     006334R         1086#   1133
MES10    006517R         1108#   1141
MES11    006541R         1112#   1144
MES12    006563R         1116#   1146
MES14    006611R         1120#   1153
MES15    006766R          437    1159#
MES2     006357R         1090#   1135
MES3     006411R         1095#   1137
MES4     006451R         1101#   1150    1155
MES5     006463R         1103#   1152    1157
MES6     006476R         1105#   1139
MIXDV    000440R          317#    403     750     782*    799*    811
MUDNAM   000000R          205#
MODSP    000252R          219     268#
MSGN$  = 104403           270#    437     501     544     728     733     861     881     884     887     926     944
MSGS$  = 104402           270#
MSG$   = 104401           270#
NOT      006734R          926    1144#
```

```
NOTRDY  005222R          402    478    511    524    548    794    901    908#
NTRUPT  004072R          710#  1082
NULL  = 000000           270#  1018
NUMB    006777R         1151   1156   1163#
OPEN  = 000000           206    212    213    214    215    232    233    234    235    236    237    238    239
                         241    243    245    246    248    249    250    253    254    256    257    259    260
                         261    262    270#
OTOA$ = 104420           270#   838
PASCNT  000034R          220#
PA18    000366R          296#   696*   701
PA22    000372R          298#   702
PICKBK  004552R          457    739    807#   816
PICKDR  004434R          379    393    459    781#
PIRQ$ = 000004           270#   718
POPSP = 005726           270#
POPSP2= 022626           270#
PORTHG  006744R          501    544   1150#
PRTY  = 000000           270#
PRTY0 = 000000           210    270#
PRTY1 = 000040           270#
PRTY2 = 000100           270#
PRTY3 = 000140           270#
PRTY4 = 000200           270#
PRTY5 = 000240           209    270#
PRTY6 = 000300           270#
PRTY7 = 000340           270#
PS    = 177776           270#
PSW   = 177776           270#
PUSH  = 005746           270#
PUSH2 = 024646           270#
RAND$ = 104417           270#
RANNUM  000054R          228#
RBUF    000444R          252    319#   406    493*   494*   514    528    530    532    551    558*   821
RBUFEA  000130R          254#   620    636    652    666
RBUFPA  000126R          253#   487    517    616    632    648    664
RBUFSZ  000132R          255#   613    629    645    661    822
RBUFVA  000124R          252#   383    453
RD00    003642R          512    525    675#   676
READ    003262R          405    486    516    553    611#   612
READY   005460R          400    476    509    522    546    792    929    959#
RESTRT  002236R          247    447#
RES1    000056R          230#   694   1072
RES2    000060R          231#
RES2X   001752R          393#   441    738    948
REZET   005650R          362   1001#  1006
RMAS    001536R          351#   687    688*   713    715*   890*  1021   1022*  1042*
RMBA    001524R          346#   584*   600*   616*   632*   648*   664*   696    702*   1032*
RMBAE   001570R          364#   703*  1078#
RMCS1   001520R          344#   497*   534*   560*   680*   683#   684    689*   708*   845    850    910*   911
                         915    933    942*   960*   961    979    996    997   1001   1009   1028*
RMCS2   001530R          348#   679*   693*   909*   922    959*   981   1007*  1036*
RMCS3   001572R          365#  1080*
RMDA    001526R          347#   585*   601*   617*   633*   649*   665*  1034*
RMDB    001542R          353#   892    895*   900*  1046*
RMDC    001554R          358#   586*   602*   618*   634*   650*   666*  1056*
RMDS    001532R          349#   964   1038*
```



```
RMDT    001546R          355#   795    797   1050*
RMEC1   001564R          362#  1064#
RMEC2   001566R          363#  1066#
RMER1   001534R          350#  1040#
RMER2   001562R          361#  1062#
RMHR    001556R          359#  1058#
RMLA    001540R          352#  1044#
RMMR1   001544R          354#  1048#
RMMR2   001560R          360#  1060#
RMOF    001552R          357#   690*  1054#
RMSN    001550R          356#  1052#
RMWC    001522R          345#   583*   599*   615*   631*   647*   663*  1030*
RSTRT   000112R          247#
RSTRT1  002256R          445    448    450    452#
S       000312R          286#   322    323    324    325    326    327    328    329    330    331    332    333
                         334    335    336    337    338    339    340    341    851*   856    863    865    867
                         869    871    879    888    890    893   934*
SBADR   000102R          240#   990#
SETUP   005770R          378   1027#
SETUP2  006232R          381   1068#
SOFCNT  000042R          223#
SOFER$= 104406           270#
SOFPAS  000046R          225#
SPOINT  000032R          219#   374    734    945
SPSIZ = 000040             1#   263
SR1     000016R          212#   385    474    730    859   1075
SR2     000020R          213#
SR3     000022R          214#
SR4     000024R          215#
START   001604R          218    370#   451
STAT    000026R          217#
SVR0    000062R          232#
SVR1    000064R          233#
SVR2    000066R          234#
SVR3    000070R          235#
SVR4    000072R          236#
SVR5    000074R          237#
SVR6    000076R          238#
SYSCNT  000052R          227#
TABLE   001444R          321#   875    898    940
TIMER   000426R          312#   496*   498*   521*   541*
TOUT    006740R         1146#
TRACK   000404R          303#   747*   760*   764    774*   778
TRERR   006710R          881   1133#
TRPDFD= 000022           270#
TRY     000436R          316#   395*   461#   725*   726
UNITNO  000414R          307#   377*   387*   455*   679    693    711    740*   783*   784    787    829    838
                         909    959
VECTOR  000010R          208#  1081
WASADR  000104R          242#   992*
WBUFEA  000136R          257#   588    604
WBUFPA  000134R          256#   584    600
WBUFRQ  000140R          258#
WBUFSZ  000142R          259#   581    597
WDFR    000116R          249#   371*
WDTO    000114R          248#   370*
```

```
WRITCB  003452R          555     643#    644
WRITCK  003166R          485     595#    596
WRITE   003072R          481     579#    580
WRITEB  003356R          554     627#    628
WRT00   003546R          495     559     659#    660
WRT00R  003562R          661#    677
XBLK    001600R          368#
XFERAD  001574R          366#
XFERCT  001576R          367#
XFLAG   000005R          206#
XMEM    000370R          297#    593#    609#    625#    641#    657#    673#    697#    698#    699#    700#    706#    707
XMES15  006651R         1126#   1159
ZERO    000430R          313#    895
.     = 007002R          319#    401     483     488     518     737     947     1132#   1162#   1165#

. ABS.  000000      000
        007002      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XRMBC0,XRMBC0/SOL/CRF:SYM=DDXCOM,XRMBC0
 RUN-TIME: 2 3 .5 SECONDS
 RUN-TIME RATIO: 27/5=4.7
 CORE USED:  7K  (13 PAGES)
```

.REM -

IDENTIFICATION

PRODUCT CODE: AC-E965G-MC

PRODUCT NAME: CXRLAGO RL11/RLV11/RLV12/RL01/RL02

PRODUCT DATE: JUNE 1982

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

RLA IS AN IOMODX THAT EXERCISES RL01/RL02 DISK DRIVES ON RL11/RLV11/RLV12 CONTROLLERS. IT EXERCISES THE DRIVES BY DOING READ HEADERS, SEEKS, READS, WRITES AND IN-CORE COMPARISONS.

ALL ERRORS DETECTED ARE REPORTED ON THE CONSOLE DEVICE.

2. REQUIREMENTS

HARDWARE: 1 TO 4 RL01/RL02 DISK DRIVES (WITH SCRATCH PACKS).
1 RL11, RLV11, OR RLV12 DISK CONTROLLER.

STORAGE:: RLA REQUIRES:
1. DECIMAL WORDS: 1445
2. OCTAL WORDS: 2645
3. OCTAL BYTES: 5512

3. PASS DEFINITION

ONE PASS OF THE RLA MODULE CONSISTS OF 20 X 100 CYCLES OF THE BASIC TEST SEQUENCE (READ HEADER, SEEK, READ HEADER, WRITE, WRITE CHECK, READ). THE TEST SEQUENCE WRITES 1024 WORDS, READS BACK THE FIRST 256, AND DATA CHECKS THE SAME.

4. EXECUTION TIME

ONE PASS OF RLA RUNNING ALONE ON A PDP-11/40 TAKES AP-PROXIMATELY ONE MINUTE.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:
DEVADR: 174400, VECTOR: 160, NR1: 5, DEVCNT: 1

6. DEVICE/OPTION SETUP

INSURE THAT ALL DRIVES UNDER TEST ARE POWERED UP, WRITE ENABLED, AND READY -- AND OF COURSE, SCRATCH PACKS INSTALLED !!!

IF MORE THAN 1 DRIVE, CHANGE UVID1: (LOC 14') ACCORDINGLY.

IF RLV12 CONTROLLER, SET SR1: = 10 (LOC 16', BIT 3).

7. SR1 OPTIONAL SETUP

BIT 0 - DROP DRIVE ON ERROR

BIT 1 - RANDOM SEEKS

BIT 2 - DON'T PRINT SOFT ERRORS

```
                BIT 3 - CONTROLLER IS AN RLV12

8.      ERROR REPORTING
        ---------------

        ON ERROR ALL REGISTERS ARE PRINTED IN THE FOLLOWING ORDER:

        RL11/RLV11      RLCS RLBA RLDA RLMP DRIVE STATUS

        RLV12           RLCS RLBA RLDA RLMP RLBAE DRIVE STATUS

*****************************************************************************
*                                                                           *
*       EDIT:   BY:             DATE:           REASON:                     *
*                                                                           *
*       1       G.PASQUINTONIO  MAY-81          "HNE" ERRORS WERE BEING     *
*                                               REPORTED AS "OPI".          *
*                                                                           *
*       2       G.PASQUINTONIO  MAY-81          CONTROLLER ERROR ON         *
*                                               WRITE-CHECK CAUSED A        *
*                                               TERMINAL TABLE SEARCH       *
*                                               (ULTIMATELY TRAPPING TO 4). *
*                                                                           *
*       3       G.PASQUINTONIO  MAY-81          "DROP ON ERROR" OPTION      *
*                                               (SR1<0>) NOT IMPLEMENTED    *
*                                               CORRECTLY.                  *
*                                                                           *
*       4       G.PASQUINTONIO  MAY-81          RLV12 22 BIT UPGRADE.       *
*                                                                           *
*       5       P. ANASTAS      DEC-81          DATA CHECK ERRORS ON FLAGGED *
*                                               BAD SPOTS ABOVE CYL. 177    *
*                                               24 DECIMAL PATCH LOCATIONS  *
*                                               ADDED.                      *
*****************************************************************************
-
```



```
000000                  IDMODX  <RLAG >,174400,160,5,0,0,20.,,146,RUFIN,256.,1024.
000000                  MODULE  150000,RLAG ,174400,160,5,0,0,20.,,146,RUFIN,256.,1024.
                        .TITLE  RLAG DEC/X11 SYSTEM EXERCISER MODULE
                ;       IDXCOM  VERSION  6.4    28-JAN-82
                                .LIST   BIN
                        ;*****************************************************************************
000000                  BEGIN:
000000  000122  000114  MODNAM: .ASCII  /RLAG / ;MODULE NAME.
000003  000107  000040
000005  000000          XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006  174400          ADDR:   174400+0                ;1ST DEVICE ADDR.
000010  000160          VECTOR: 160+0                   ;1ST DEVICE VECTOR.
000012  000240          BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013  000000          BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014  000001          DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016  000000          SR1:    OPEN                    ;SWITCH REGISTER 1
000020  000000          SR2:    OPEN                    ;SWITCH REGISTER 2
000022  000000          SR3:    OPEN                    ;SWITCH REGISTER 3
000024  000000          SR4:    OPEN                    ;SWITCH REGISTER 4
                        ;*****************************************************************************
000026  150000          STAT:   150000                  ;STATUS WORD.
000030  000252'         INIT:   START                   ;MODULE START ADDR.
000032  000252'         SPOINT: MODSP                   ;MODULE STACK POINTER.
000034  000000          PASCNT: 0                       ;PASS COUNTER.
000036  000024          ICONT:  20.                     ;# OF ITERATIONS PER PASS=20.
000040  000000          ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042  000000          SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044  000000          HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046  000000          SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050  000000          HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052  000000          SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054  000000          RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                  CONFIG:                         ;RESERVED FOR MONITOR USE
000056  000000          RES1:   0                       ;RESERVED FOR MONITOR USE
000060  000000          RES2:   0                       ;RESERVED FOR MONITOR USE
000062  000000          SVR0:   OPEN                    ;LOC TO SAVE R0.
000064  000000          SVR1:   OPEN                    ;LOC TO SAVE R1.
000066  000000          SVR2:   OPEN                    ;LOC TO SAVE R2.
000070  000000          SVR3:   OPEN                    ;LOC TO SAVE R3.
000072  000000          SVR4:   OPEN                    ;LOC TO SAVE R4.
000074  000000          SVR5:   OPEN                    ;LOC TO SAVE R5.
000076  000000          SVR6:   OPEN                    ;LOC TO SAVE R6.
000100  000000          CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102                  SBADR:                          ;ADDR OF GOOD DATA, OR
000102  000000          ACSR:   OPEN                    ;CONTENTS OF CSR.
000104                  WASADR:                         ;ADDR OF BAD DATA, OR
000104  000000          ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106                  ERRTYP:                         ;TYPE OF ERROR
000106  000000          ASB:    OPEN                    ;EXPECTED DATA.
000110  000000          AWAS:   OPEN                    ;ACTUAL DATA.
000112  000424'         RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114  000000          WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116  000000          WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120  000000          INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122  000146          IDNUM:  146                     ;MODULE IDENTIFICATION NUMBER=146
000124  003636'         RBUFVA: BUFIN                   ;READ BUFFER VIRTUAL ADDRESS
000126  000000          RBUFPA: OPEN                    ;READ BUFFER PHYSICAL ADDRESS
```

DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
000130  000000                    RBUFEA: OPEN                    ;READ BUFFER EA BITS
000132  000400                    RBUFSZ: 256.                    ;SIZE OF THE READ BUFFER
000134  000000                    WBUFPA: OPEN                    ;WRITE BUFFER PHYSICAL ADDRESS
000136  000000                    WBUFEA: OPEN                    ;WRITE BUFFER EA BITS
000140  002000                    WBUFRQ: 1024.                   ;WRITE BUFFER SIZE REQUESTED
000142  000000                    WBUFSZ: OPEN                    ;WRITE BUFFER SIZE AVAILABLE
000144  000000                    CDERCT: OPEN                    ;CDATA/DATCK ERROR COUNT
000146  000000                    CDWDCT: OPEN                    ;CDATA/DATCK WORD COUNT
000150  000000                    FREE:   OPEN                    ;RESERVED FOR FUTURE USE
        000040                            .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                          .NLIST
                                          .WORD   0
                                          .LIST
                                          .ENDR
        000252                    MODSP:
                                  ;*****************************************************************************
    159
    160 000252  012767  002000  177636  START:  MOV     #1024.,WDFR     ;1024. WORDS FROM MEM/ITERATION
    161 000260  012767  000400  177626          MOV     #256.,WDTO      ;256 WORDS TO MEM/ITERATION
    162 000266  012767  000005  177624          MOV     #5,INTR         ;5 INTERRUPTS/ITERATION
    163 000274  012767  000057  004424          MOV     #57,NUMB
    164 000302  012767  177400  003316          MOV     #-400,DRIVE     ;SET DRIVE SELECT
    165 000310  012767  000001  003254          MOV     #1,DRVMSK       ;SETUP DRIVE SELECT MASK
    166 000316  005067  003274                  CLR     DLTCNT          ;CLEAR DATA LATE COUNT
    167 000322  004767  001300                  JSR     PC,SETUP        ;GO SET UP REGISTERS
    168 000326  016767  177462  003274          MOV     DVID1,DVICE     ;COPY DRIVE SELECTION
    169 000334  122737  000014  000041  3$:     CMPB    #14,@#41        ;WAS RL LOAD DEVICE?
    170 000342  001020                          BNE     7$              ;N=BRANCH; Y=SEE IF LOAD UNIT SELECTED
    171 000344  012702  000001                  MOV     #1,R2           ;SET UP FOR MASK
    172 000350  113701  000040                  MOVB    @#40,R1         ;GET LOAD UNIT
    173 000354  001403                          BEQ     5$              ;IF ZERO GO MASK OUT UNIT
    174 000356  006302                  4$:     ASL     R2              ;SHIFT MASK
    175 000360  105301                          DECB    R1              ;DEC COUNT
    176 000362  001375                          BNE     4$              ;KEEP CHECKING
    177 000364  030267  003240          5$:     BIT     R2,DVICE        ;WAS THAT DRIVE SELECTED?
    178 000370  001405                          BEQ     7$              ;N=BRANCH; Y=CONTINUE
    179 000372  040267  003232                  BIC     R2,DVICE        ;DELETE UNIT FROM DEVICE MAP
    180 000376  104403  000000' 005474'        MSGN$,BEGIN,DROPLD      ;ASCII MESSAGE CALL WITH COMMON HEADER
    181 000404  005767  003220          7$:     TST     DVICE           ;ANY DRIVES SELECTED?
    182 000410  001011                          BNE     RSTRT1          ;YES, CONTINUE
    183 000412  104403  000000' 005424'        MSGN$,BEGIN,ABORT       ;ASCII MESSAGE CALL WITH COMMON HEADER
    184 000420  000167  001176                  JMP     FIN1            ;MESSAGE, DROP MODULE
    185 000424  005767  003162          RESTRT: TST     CNT             ;+    / SUPPORT
    186 000430  001001                          BNE     RSTRT1          ;+   /  FOR
    187 000432  000707                          BR      START           ;+  /   DT03
    188 000434                          RSTRT1:                         ;+/
    189 000434  104415  000000' 000124'        GETPA$,BEGIN,RBUFVA             ;GET PHYSICAL ADDRESS FROM 16-BIT RBUFVA
    190 000442  000421                          BR      CHKDR1
    191 000444  006367  003122          LOOPL:  ASL     DRVMSK          ;SHIFT MASK FOR NEXT DRIVE
    192 000450  022767  000020  003114          CMP     #20,DRVMSK      ;DRIVE MASK OVERSHIFT CHECK
    193 000456  001013                          BNE     CHKDR1          ;BRANCH IF MASK OK
    194 000460  012767  000001  003104          MOV     #1,DRVMSK       ;RESET DRIVE SELECT MASK
    195 000466  012767  000057  004232          MOV     #57,NUMB
    196 000474  012767  177400  003124          MOV     #-400,DRIVE     ;RESET DRIVE SELECT
    197 000502                          CHKDRV:
        000502  104413  000000'                ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                                        ;MONITOR SHALL TEST END OF PASS
```


DEC/X11 SYSTEM EXERSIZER MACRO DEFINITION MODULE

```
    198 000506                          CHKDR1:
    199 000506  062767  000400  003112  1$:     ADD     #400,DRIVE      ;NEXT DRIVE
    200 000514  005267  004206                  INC     NUMB
    201 000520  036767  003046  003102          BIT     DRVMSK,DVICE    ;IS THAT DRIVE PRESENT
    202 000526  001746                          BEQ     LOOPL           ;NO, GO FOR NEXT ONE
    203 000530  005067  002772                  CLR     RETRY           ;CLEAR A FEW LOCATIONS
    204 000534  005067  003026                  CLR     RWER            ;READ WRITE ERROR FLAG
    205 000540  005067  003046                  CLR     CNT             ;COUNT
    206                                 ;
    207                                 ;WE HAVE A DRIVE, START TESTING
    208                                 ;
    209 000544  004767  000722                  JSR     PC,WTRDY
    210 000550  004567  001430                  JSR     R5,DRVRTS       ;ISSUE DRIVE RESET, CLEAR VOLUME
    211 000554  012767  100077  003024          MOV     #100077,MASK2   ;
    212 000562  012767  077724  003004          MOV     #77724,FSTBSC   ;SET UP FOR RL01
    213 000570  012767  077750  003000          MOV     #77750,LSTBSC
    214 000576  012767  100177  002774          MOV     #100177,MASK
    215 000604  012767  077600  002770          MOV     #77600,LSTCYL
    216 000612  012767  077700  002764          MOV     #77700,LSTTRK
    217 000620  032767  000200  002730          BIT     #200,T.MP       ;TEST RL01 OR RL02
    218 000626  001422                          BEQ     2$              ;RL01 BRANCH
    219 000630  042767  100000  002742          BIC     #100000,MASK    ;FIX FOR RL02
    220 000636  052767  100000  002736          BIS     #100000,LSTCYL
    221 000644  052767  100000  002732          BIS     #100000,LSTTRK
    222 000652  052767  100000  002716          BIS     #100000,LSTBSC
    223 000660  042767  100000  002720          BIC     #100000,MASK2   ;
    224 000666  052767  100000  002700          BIS     #100000,FSTBSC
    225 000674  012767  000201  002722  2$:     MOV     #201,DIFWD
    226 000702  004567  001256          LOOP:   JSR     R5,RDHDR        ;READ HEADER ON DISK
    227 000706  016767  002644  002706          MOV     T.MP,HDRWD      ;GET HEADER
    228                                 ;
    229                                 ;CHECK TO SEE IF RANDOM SEEK IS REQUESTED, BIT 1 OF SR1
    230                                 ;SET INDICATES A RANDOM SEEK OTHERWISE SEEK IS INCREMENTAL
    231                                 ;
    232 000714  032767  000002  177074  TAG:    BIT     #BIT1,SR1       ;INCREMENTAL OR RANDOM SEEKS?
    233 000722  001446                          BEQ     TAG1            ;INCREMENTAL, TAG1
    234 000724  042767  000177  002670          BIC     #177,HDRWD      ;CLEAR HEAD AND SECTOR BITS
    235 000732  104417  000000'                RAND$,BEGIN
    236 000736  016700  177112                  MOV     RANNUM,R0       ;STORE IT AWAY
    237 000742  010001                          MOV     R0,R1           ;SAVE A COPY
    238 000744  046700  002630                  BIC     MASK,R0         ;CLEAR HEAD AND SECTOR
    239 000750  010067  002659                  MOV     R0,DIFWD        ;LET'S CALCULATE DIFFERENCE WORD
    240 000754  166767  002642  002642          SUB     HDRWD,DIFWD     ;GET DIFFERENCE TO SEEK
    241 000762  103003                          BCC     1$
    242 000764  005467  002634                  NEG     DIFWD           ;MAKE DIFF ABSOLUTE
    243 000770  000403                          BR      2$
    244 000772  052767  000004  002624  1$:     BIS     #4,DIFWD        ;SET DIRECTION BIT
    245 001000  052767  000001  002616  2$:     BIS     #1,DIFWD        ;SET MARKER
    246 001006  032701  000100                  BIT     #100,R1         ;TEST HEAD
    247 001012  001403                          BEQ     3$              ;IF 0, DON'T SET HEAD IN DIFF
    248 001014  052767  000020  002602          BIS     #20,DIFWD       ;SET HEAD
    249 001022  010167  002574          3$:     MOV     R1,HDRWD        ;GET EXPECTED HEADER
    250 001026  046767  002554  002566          BIC     MASK2,HDRWD     ;CLEAR SECTOR BITS
    251 001034  000167  000134                  JMP     TAG2
    252
    253 001040  042767  000177  002554  TAG1:   BIC     #177,HDRWD      ;CLEAR OUT SECTOR BITS & HEAD
    254 001046  032767  177600  002546          BIT     #177600,HDRWD   ;ON TRACK 0?
```

```
 255 001054  001007                          BNE     1$              ;NO, GO CHECK FOR LAST CYLINDER
 256 001056  012767  000200  002536          MOV     #200,HDRWD      ;SET NEXT ADDRESS=CYL 1
 257 001064  012767  000205  002532          MOV     #205,DIFWD      ;DIF WD 1, MARKER, SEEK IN, HS=0
 258                                                                 ;SET CURRENT HD=0, SEEK IN
 259 001072  000440                          BR      TAG2
 260 001074  026767  002502  002520  1$:     CMP     LSTCYL,HDRWD    ;CURRENT ADDRESS=LAST TRACK?
 261 001102  001012                          BNE     2$              ;NO, CONTINUE
 262 001104  162767  000200  002510          SUB     #200,HDRWD      ;
 263 001112  052767  000100  002502          BIS     #100,HDRWD      ;
 264 001120  012767  000221  002476          MOV     #221,DIFWD      ;DIF WD 1, MARKER, SEEK OUT, HS=1
 265                                                                 ;SET CURRENT HD=1, SEEK OUT
 266 001126  000422                          BR      TAG2
 267 001130  032767  000004  002466  2$:     BIT     #4,DIFWD        ;SN SET IN DIF WORD
 268 001136  001404                          BEQ     3$              ;NO, 3$
 269 001140  062767  000200  002454          ADD     #200,HDRWD      ;YES, CYL WILL INCREMENT
 270 001146  000403                          BR      4$              ;SKIP OVER
 271 001150  162767  000200  002444  3$:     SUB     #200,HDRWD      ;NO, CYL WILL DECREMENT
 272 001156  032767  000020  002440  4$:     BIT     #20,DIFWD       ;HEAD SET?
 273 001164  001403                          BEQ     TAG2            ;NO, LEAVE EXPECTED ALONE
 274 001166  052767  000100  002426          BIS     #100,HDRWD      ;YES, SET HEAD SELECT BIT
 275 001174  004567  000740          TAG2:   JSR     R5,SEEK         ;PERFORM SEEK
 276 001200  004767  000266                  JSR     PC,WTRDY        ;WAIT FOR SEEK TO FINISH
 277 001204  004567  000754                  JSR     R5,RDHDR        ;READ HEADER VERIFY CORRECT
 278                                                                 ;SEEK
 279 001210  016767  002342  003504          MOV     T.MP,CURADR     ;READ HEADER
 280 001216  042767  000077  003476          BIC     #77,CURADR      ;CLEAR OUT SECTOR BITS
 281 001224  026767  003472  002370          CMP     CURADR,HDRWD    ;WAS SEEK CORRECT?
 282 001232  001425                          BEQ     6$              ;YES, CONTINUE
 283                                                                 ;NO REPORT ERROR
 284 001234  016767  002272  176636          MOV     RLCS,CSRA
 285 001242  017767  002264  176632          MOV     @RLCS,ACSR
 286 001250  017767  002256  176626          MOV     @RLCS,ASTAT
 287 001256  104403  000000' 005470'         MSGNS,BEGIN,BDSEEK      ;ASCII MESSAGE CALL WITH COMMON HEADER
 288
 289 001264  012767  000051  176614          MOV     #51,ERRTYP      ;BAD SEEK
 290                                         ;****************************************************************
     001272  104405  000000' 005500'         HRDERS,BEGIN,TABLE      ;SEEK WAS BAD
                                             ;****************************************************************
 291
 292 001300  016767  003416  002314          MOV     CURADR,HDRWD    ;MAKE MISTAKE NEW HDRWD
 293
 294 001306  026767  002310  002270  6$:     CMP     HDRWD,LSTTRK    ;ARE WE ON LAST TRACK
 295 001314  001002                          BNE     7$              ;NO, CONTINUE
 296 001316  000167  177372                  JMP     TAG             ;YES, GO GET ANOTHER CAUSE ITS THE BAD SECTOR TRACK
 297
 298 001322  016767  176604  002304  7$:     MOV     RBUFSZ,WCNT2    ;GET BUFFER SIZE (READ)
 299 001330  005467  002300                  NEG     WCNT2           ;NEGATE FOR RLMP
 300 001334  104414  000000'                 GWBUFS, BEGIN           ;GET WRITE BUFFER INFORMATION
 301 001340  016767  176576  002264          MOV     WBUFSZ,WCNT1    ;GET BUFFER SIZE (WRITE)
 302 001346  005467  002260                  NEG     WCNT1           ;NEGATE FOR RLMP
 303 001352  004567  000502                  JSR     R5,WRITE        ;WRITE DATA
 304 001356  005767  002204                  TST     RWER            ; CONTROLLER ERROR ??
 305 001362  001017                          BNE     5$              ; SKIPS READS IF SO.
 306 001364  004567  000460                  JSR     R5,WRCHK
 307 001370  005767  002172                  TST     RWER            ; ERROR ??
 308 001374  001012                          BNE     5$              ; SKIP READS IF SO.
 309 001376  004567  000506                  JSR     R5,READ         ;READ DATA
```



```
 310 001402  005767  002160                  TST     RWER            ; ERROR ??
 311 001406  001005                          BNE     5$              ; SKIP DATA CHECK IF SO.
 312 001410  104412  000000' 000126'         CDATA$,BEGIN,RBUFPA     ; REQUEST FOR MONITOR TO CHECK DATA
     001416  001422'                         5$                      ; IF ERROR, RETURN AT TAG 5$
 313 001420  000410                          BR      4$              ; BR IF NO DATA ERRORS.                 ;GP-3
 314 001422  032767  000001  176366  5$:     BIT     #BIT0,SR1       ; DROP ON ERROR ??                      ;GP-3
 315 001430  001404                          BEQ     4$              ; NO                                    ;GP-3
 316 001432  004567  000134                  JSR     R5,DROP         ; YES, DROP IT.                         ;GP-3
 317 001436  000167  177002                  JMP     LOOPL           ;...AND TRY ANOTHER.                    ;GP-3
 318
 319 001442  005067  002120          4$:     CLR     RWER
 320 001446  005267  002140                  INC     CNT
 321 001452  022767  000020  002132          CMP     #16.,CNT
 322 001460  001402                          BEQ     3$
 323 001462  000167  177214                  JMP     LOOP            ; REITERATE 16 TIMES/DRIVE...
 324 001466  000167  176752          3$:     JMP     LOOPL           ;...THEN GET ANOTHER DRIVE.
 325                                         ;
 326                                         ;WAIT FOR DRIVE READY.  DROP DRIVE IF IT NEVER COMES UP.
 327                                         ;
 328 001472  042777  001400  002032  WTRDY:  BIC     #1400,@RLCS
 329 001500  056777  002122  002024          BIS     DRIVE,@RLCS
 330 001506  012767  077777  002074          MOV     #77777,CLK      ;SET UP TIMEOUT
 331 001514  032777  000001  002010  1$:     BIT     #1,@RLCS        ;DRIVE READY?
 332 001522  001401                          BEQ     2$              ; NOT YET                               ;GP-3
 333 001524  000207                          RTS     PC              ; YES, RETURN TO CALLER.                ;GP-3
 334 001526                          2$:
     001526  104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
     001532  104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 335 001536  005367  002046                  DEC     CLK             ;CHECK TIMEOUT
 336 001542  001364                          BNE     1$              ; LOOP TIL TIMER EXPIRES.
 337 001544  012767  000006  176334          MOV     #6,ERRTYP       ;DRIVE NOT READY
 338                                         ;****************************************************************
     001552  104405  000000' 003532'         HRDERS,BEGIN,RLCS       ;DRIVE NOT READY
                                             ;****************************************************************
 339 001560  004567  000006                  JSR     R5,DROP         ;CLEAR DRIVE (DROP) FROM LIST.
 340 001564  005726                          TST     (SP)+           ; FIX THE STACK...
 341 001566  000167  176652                  JMP     LOOPL           ;...AND GO FOR ANOTHER.
 342                                         ;
 343                                         ;DROP CURRENT DRIVE.  DROP MODULE IF NO DRIVES LEFT.
 344                                         ;
 345 001572                          DROP:
     001572  104403  000000' 005410'         MSGNS,BEGIN,DRUPMS      ;ASCII MESSAGE CALL WITH COMMON HEADER
 346 001600  046767  001766  002022          BIC     DRVMSK,DVICE    ;CLEAR THIS DRIVE BIT.
 347 001606  001401                          BEQ     1$              ; DROP MODULE IF NO MORE.               ;GP-3
 348 001610  000205                          RTS     R5              ; OTHERWISE, RETURN TO CALLER.          ;GP-3
 349 001612                          1$:
     001612  104403  000000' 005420'         MSGNS,BEGIN,NOLEFT      ;ASCII MESSAGE CALL WITH COMMON HEADER
 350 001620  005726                          TST     (SP)+           ; FIX STACK.
 351 001622                          FIN1:
     001622  104410  000000'                 ENDS,BEGIN              ;DROP THE MODULE
 352                                         ;
 353                                         ;ROUTINE TO SET UP RL11 REGISTERS, VECTOR AND BR LEVEL
 354                                         ;
 355 001626  016700  176154          SETUP:  MOV     ADDR,R0         ;GET BASE ADDRESS
 356 001632  010067  001674                  MOV     R0,RLCS         ;CONTROL REGISTER
 357 001636  005720                          TST     (R0)+           ;INCREMENT FOR NEXT
 358 001640  010067  001670                  MOV     R0,RLBA         ;BUS ADDRESS
```

```
359 001644  005720                          TST     (R0)+           ;INCREMENT FOR NEXT
360 001646  010067  001664                  MOV     R0,RLDA         ;DISK ADDRESS
361 001652  005720                          TST     (R0)+           ;INCREMENT FOR NEXT
362 001654  010067  001660                  MOV     R0,RLMP         ;DATA BUFFER
363 001660  032767  000010  176130          BIT     #BIT3,SR1       ;TEST IF AN RLV12 CONTROLER                     ;GP-4
364 001666  001411                          BEQ     1$              ;BR IF NOT                                      ;GP-4
365 001670  005720                          TST     (R0)+           ;INCREMENT FOR NEXT                             ;GP-4
366 001672  010067  001644                  MOV     R0,RLBAE        ;BUS EXTENDED ADDRESS REGISTER                  ;GP-4
367 001676  012767  003560' 003604          MOV     #T.BAE,TABLEY   ;FIX ERROR REPORT                               ;GP-4
368 001704  012767  003562' 003600          MOV     #T.STAT,TABLEZ  ; FOR RLV12 CONTROLLER                          ;GP-4
369 001712  016700  176072          1$:     MOV     VECTOR,R0       ;GET VECTOR ADDRESS
370 001716  012720  000252'                 MOV     #STAPT,(R0)+    ;SET POINTER
371 001722  116710  176064                  MOVB    #R1,(R0)        ;SET PRIORITY
372 001726  000207                          RTS     PC              ;RETURN
373                                 ;
374                                 ; SUBROUTINE TO SET 18 OR 22 BIT BUS ADDRESS IN CONTROLLER.
375                                 ; THIS ENTIRE SUBROUTINE IS NEW.                                                ;GP-4
376                                 ;
377 001730  016767  176200  000102  SETWBA: MOV     WBUFPA,PA18     ; GET 18 BIT WRITE BUFFER ADDRESS.
378 001736  016767  176174  000076          MOV     WBUFEA,EA18
379 001744  000406                          BR      SETCMN
380 001746  016767  176154  000064  SETRBA: MOV     RBUFPA,PA18     ; GET 18 BIT READ READ BUFFER ADDRESS.
381 001754  016767  176150  000060          MOV     RBUFEA,EA18
382 001762  016777  000052  001544  SETCMN: MOV     PA18,@RLBA      ; SET BA<15:0>...
383 001770  042777  000060  001534          BIC     #60,@RLCS
384 001776  056777  000040  001526          BIS     EA18,@RLCS      ;...AND BA<17:16> IN CSR<5:4>.
385 002004  032767  000010  176004          BIT     #BIT3,SR1       ; ARE WE RLV12 ??
386 002012  001411                          BEQ     1$              ; BR IF NOT.
387 002014  104416  000000' 002040'        .MAP22$, BEGIN,PA18     ; GET 22-BIT ADDR FROM 18-BIT ADDR
388 002022  016777  000016  001504          MOV     PA22,@RLBA      ; LOAD BA<15:0>...
389 002030  016777  000012  001504          MOV     EA22,@RLBAE     ;...AND BA<21:16>.
390 002036  000207                  1$:     RTS     PC
391
392 002040  000000                  PA18:   0               ; 18 BIT BA<15:0>...
393 002042  000000                  EA18:   0               ;...AND EA<17:16> IN BITS <5:4>.
394 002044  000000                  PA22:   0               ; 22 BIT BA<15:0>...
395 002046  000000                  EA22:   0               ;...AND EA<21:16> IN BITS <5:0>.
396                                 ;
397                                 ;DRIVERS (INTERRUPT)
398                                 ;
399 002050  012767  000102  001542  WRCHK:  MOV     #102,FUNC       ; WRITE-CHECK...
400 002056  000403                          403                     ;...OR...
401 002060  012767  000112  001532  WRITE:  MOV     #112,FUNC       ;...WRITE FUNCTION.
402 002066  016777  001540  001444          MOV     WCNT1,@RLMP     ; WORD COUNT
403 002074  016777  001522  001434          MOV     HDRWD,@RLDA     ; DISK ADDRESS
404 002102  004767  177622                  JSR     PC,SETWBA       ; WRITE BUFFER ADDRESS                          ;GP-4
405 002106  000444                          BR      EXEC
406 002110  012767  000114  001502  READ:   MOV     #114,FUNC       ; READ FUNCTION
407 002116  016777  001512  001414          MOV     WCNT2,@RLMP     ; WORD COUNT
408 002124  016777  001472  001404          MOV     HDRWD,@RLDA     ; DISK ADDRESS
409 002132  004767  177610                  JSR     PC,SETRBA       ; READ BUFFER ADDRESS                           ;GP-4
410 002136  000430                          BR      EXEC
411 002140  012767  000106  001452  SEEK:   MOV     #106,FUNC       ;SEEK FUNCTION
412 002146  016777  001452  001362          MOV     DIFWD,@RLDA     ;DIFFERENCE WORD
413 002154  052777  000001  001354          BIS     #1,@RLDA        ;SET MARKER BIT
414 002162  000416                          BR      EXEC
415 002164  012767  000110  001426  RDHDR:  MOV     #110,FUNC       ;READ HEADER FUNCTION
```



```
416 002172  000412                          BR      EXEC
417 002174  012777  000003  001334  GSTAT:  MOV     #3,@RLDA        ;GET STATUS...
418 002202  000403                          403                     ;...OR...
419 002204  012777  000013  001324  DRVRTS: MOV     #13,@RLDA       ;...RESET AND GET STATUS.
420 002212  012767  000104  001400          MOV     #104,FUNC       ;GET STATUS FUNCTION.
421
422 002220  042777  001416  001304  EXEC:   BIC     #1416,@RLCS     ; CLEAR THE OLD...                              ;GP-4
423 002226  056777  001374  001276          BIS     DRIVE,@RLCS     ;...AND INSERT NEW DRIVE...                     ;GP-4
424 002234  056777  001360  001270          BIS     FUNC,@RLCS      ;...AND OPCODE BITS.                            ;GP-4
425 002242  012777  002262' 175540          MOV     #INTSRV,@VECTOR ;SET UP INTERRUPT VECTOR...                     ;GP-4
426 002250  042777  000200  001254          BIC     #200,@RLCS      ;...AND EXECUTE.                                ;GP-4
427 002256  104400  000000'                  EXITS,BEGIN             ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
428                                 ;
429                                 ; CONTINUE HERE ON RL INTERRUPT.
430                                 ;
431 002262                          INTSRV:
                                    ;-----------------------------------------------------------------------
    002262  000004  000000' 002270'        PIRQ$,BEGIN,1$          ; QUEUE UP TO CONTINUE AT 1$ AND RTI
                                    ;-----------------------------------------------------------------------
432 002270  005067  001266          1$:     CLR     T.STAT
433 002274  016767  001232  175576          MOV     RLCS,CSRA       ;LOAD ADDR OF CSR
434 002302  017767  001224  175572          MOV     @RLCS,ACSR      ;LOAD CONTENTS OF CSR
435 002310  016767  175566  001232          MOV     ACSR,T.CS
436 002316  016767  001212  001226          MOV     @RLBAE,T.BA
437 002324  017767  001206  001222          MOV     @RLDA,T.DA
438 002332  017767  001202  001216          MOV     @RLMP,T.MP
439 002340  032767  000010  175450          BIT     #BIT3,SR1       ;TEST IF RLV12 CONTROLLER                       ;GP-4
440 002346  001403                          BEQ     20$             ;BR IF NOT                                      ;GP-4
441 002350  017767  001166  001202          MOV     @RLBAE,T.BAE    ;GET EA BITS                                    ;GP-4
442 002356  005767  001166          20$:    TST     T.CS            ;ANY ERRORS
443 002362  100403                          BMI     11$             ;YES, CONTINUE TO CHECK
444 002364  005067  001136                  CLR     RETRY
445 002370  000205                          RTS     R5              ;NO, RETURN CALL+4 SKIP RETRY
446
447 002372  005267  001170          11$:    INC     RWEN
448 002376  012767  005404' 003056          MOV     #NULLX,HTYPE    ;SETUP FOR NULL PRINT
449 002404  032767  040000  001136          BIT     #BIT14,T.CS     ;DRIVE ERROR
450 002412  001457                          BEQ     2$              ;NO,BRANCH
451 002414  012777  000003  001114          MOV     #3,@RLDA        ;GET STATUS
452 002422  012767  000004  001116          MOV     #4,TMP
453 002430  056767  001172  001110          BIS     DRIVE,TMP
454 002436  016777  001104  001066          MOV     TMP,@RLCS
455 002444                          99$:
    002444  104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
    002450  104407  000000'                  BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
456 002454  032777  000200  001050          BIT     #200,@RLCS
457 002462  001770                          BEQ     99$
458 002464  017767  001050  001070          MOV     @RLMP,T.STAT
459 002472  104403  000000' 005454'        MSG$,BEGIN,DRVERR       ;ASCII MESSAGE CALL WITH COMMON HEADER
460 002500  012767  000006  175400          MOV     #6,ERRTYP       ;DRIVE ERROR
461                                 ;***********************************************************************
    002506  104405  000000' 005500'        HRDER$,BEGIN,TABLE      ;
                                    ;***********************************************************************
462 002514  012777  000013  001014          MOV     #13,@RLDA
463 002522  016777  001020  001002          MOV     TMP,@RLCS
464 002530                          98$:
    002530  104407  000000'                  BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
```

```
     002534  104407  000000'                 $BREAK$,$BEGIN              ;THEN CONTINUE AT NEXT INSTRUCTION.
 465 002540  032777  000200  000764          BIT     #200,@RLCS
 466 002546  001770                          BEQ     9$
 467 002550  000522                          BR      NORPT
 468 002552  032767  020000  000770  2$:     BIT     #BIT13,T.CS         ;NXM SET
 469 002560  001404                          BEQ     21$
 470 002562  012767  005400' 002672          MOV     #NXM,HTYPE
 471 002570  000515                          BR      HRDRPT
 472 002572  032767  002000  000750  21$:    BIT     #BIT10,T.CS         ;OPI SET
 473 002600  001423                          BEQ     4$                  ;NO, CHECK DCRC,DCK
 474 002602  012767  005353' 002652          MOV     #OPI,HTYPE          ;INITIAL SET FOR OPI
 475 002610  032767  004000  000732          BIT     #BIT11,T.CS         ;HCRC?
 476 002616  001404                          BEQ     3$                  ;NO, BRANCH
 477 002620  012767  005373' 002634          MOV     #HCRC,HTYPE         ;HCRC ERROR
 478 002626  000505                          BR      FNDBSC              ;FIND BAD SECTOR
 479 002630  032767  010000  000712  3$:     BIT     #BIT12,T.CS         ;HNF
 480 002636  001472                          BEQ     HRDRPT              ; NO, REPORT AS REPORT "OPI"                ;GP-1
 481 002640  012767  005367' 002614          MOV     #HNF,HTYPE          ;HNF ERROR
 482 002646  000475                          BR      FNDBSC              ;GO CHECK BAD SECTOR FILE
 483 002650  032767  004000  000672  4$:     BIT     #BIT11,T.CS         ;DCK?
 484 002656  001406                          BEQ     5$                  ;NO,MUST BE DLT
 485 002660  005367  000670                  DEC     T.DA                ;BACK UP TO SECTOR THAT WAS BAD
 486 002664  012767  005363' 002570          MOV     #DCK,HTYPE          ;DCK ERROR
 487 002672  000463                          BR      FNDBSC              ;GO CHECK BAD SECTOR FILE
 488 002674  012767  005357' 002560  5$:     MOV     #DLT,HTYPE          ;SETUP DLT ERROR
 489
 490 002702  032767  000004  175106  RPTERR: BIT     #BIT2,SR1           ;PRINTING SOFTERRORS ??
 491 002710  001011                          BNE     55$                 ;NO, SKIP PRINT
 492 002712  104403  000000' 005460'         MSGN$,BEGIN,SOFT            ;ASCII MESSAGE CALL WITH COMMON HEADER
 493 002720  012767  000001  175160          MOV     #1,ERRTYP           ;DATA ERROR
 494                                         ;*****************************************************************
     002726  104406  000000' 005500'         SOFER$,BEGIN,TABLE          ;
                                             ;*****************************************************************
 495 002734  026767  000566  000566  55$:    CMP     RETRY,LIMIT         ;RETRY EXHAUSTED
 496 002742  001405                          BEQ     6$                  ;YES, NO MORE RETRIES
 497 002744  005267  000556                  INC     RETRY
 498 002750  162705  000004                  SUB     #4,R5               ; ADJUST RETURN PC...
 499 002754  000205                          RTS     R5                  ;...AND TRY AGAIN.
 500
 501 002756  016700  000636          6$:     MOV     FUNC,R0             ; GET FUNCTION CODE...                      ;GP-2
 502 002762  042700  177761                  BIC     #^C16,R0            ;...STRIP IT...                             ;GP-2
 503 002766  016067  004730' 002434          MOV     FNCLST(R0),EXCEED   ;...AND GET APPORPRIATE TEXT.               ;GP-2
 504 002774  016767  002462  002440          MOV     HTYPE,TER1          ;GET ERROR TYPE
 505 003002  104403  000000' 005440'         MSGN$,BEGIN,HARD            ;ASCII MESSAGE CALL WITH COMMON HEADER
 506 003010  104403  000000' 005430'         MSGN$,BEGIN,EXCEED          ;ASCII MESSAGE CALL WITH COMMON HEADER
 507 003016  005067  000504          NORPT:  CLR     RETRY
 508 003022  000205                          RTS     R5
 509 003024  016767  002432  002410  HRDRPT: MOV     HTYPE,TER1
 510 003032  104403  000000' 005440'         MSGN$,BEGIN,HARD            ;ASCII MESSAGE CALL WITH COMMON HEADER
 511 003040  000766                          BR      NORPT
 512
 513                                         ;ERROR WAS HCRC OR HNF OR DCK, POSITION TO LAST TRACK AND RECOVER
 514                                         ;BAD SECTOR FILES. IF DCK/HNF CHECK WHOLE DA, IF HCRC CHECK IF
 515                                         ;WE WERE DOING A RDHDR IF READ HDR THEN CHECK ONLY TRACK AND
 516                                         ;CYLINDER.
 517                                         ;
 518 003042  016701  000534          FNDBSC: MOV     LSTCYL,R1           ;LAST TRACK
```



```
 519 003046  016700  000550                  MOV     HDRWD,R0            ;PRESENT POSITION
 520 003052  042700  000100                  BIC     #100,R0             ;CLEAR OUT HEAD
 521 003056  160001                          SUB     R0,R1               ;CALC SEEK DIFFERENCE
 522 003060  010177  000452                  MOV     R1,@RLDA            ;LOAD SEEK DIFFERENCE
 523 003064  052777  000025  000444          BIS     #25,@RLDA           ;SET HEAD 1, SEEK IN
 524 003072  016767  000530  000464          MOV     DRIVE,MFLG          ;SELECT DRIVE. (MFLG USED)
 525 003100  052767  000006  000456          BIS     #6,MFLG             ;SET UP SEEK
 526 003106  016777  000452  000416          MOV     MFLG,@RLCS          ;SEEK
 527 003114  004767  176352                  JSR     PC,WTRDY            ;WAIT FOR SEEK TO FINISH
 528
 529                                         ;NOW SITTING ON LAST TRACK, RECOVER BAD SECTOR FILES AND COMPARE
 530
 531 003120  016700  000460                  MOV     LSTTRK,R0           ;STARTING SECTOR 0
 532 003124  005067  000434                  CLR     MFLG                ;SWITCH TO TELL US MANUF OR FIELD FILE
 533 003130  005067  000434                  CLR     FND                 ;FLAG TO INDICATE HEADER FOUND IN LIST
 534 003134  010077  000376          2$:     MOV     R0,@RLDA            ;LOAD SECTOR TO READ
 535 003140  012777  177400  000372          MOV     #-256.,@RLMP        ;TWO SECTOR READ
 536 003146  012777  000215  000356          MOV     #215,@RLCS          ; READ COMMAND                              ;GP-4
 537 003154  004767  176566                  JSR     PC,SETRBA           ; BUFFER ADDRESS.                           ;GP-4
 538 003160  056777  000432  000344          BIS     DRIVE,@RLCS         ; INSERT DRIVE                              ;GP-4
 539 003166  042777  000200  000336          BIC     #200,@RLCS          ; EXECUTE IT.
 540 003174  004767  176272                  JSR     PC,WTRDY            ;WAIT FOR DRIVE
 541 003200  005777  000326                  TST     @RLCS               ;READ SUCCESSFUL??
 542 003204  100023                          BPL     4$                  ;YES, GO CHECK FOR SECTOR
 543
 544 003206  062700  000004                  ADD     #4,R0               ;NO, NEXT SECTOR
 545 003212  005767  000346                  TST     MFLG                ;WHICH AE READING, MANUF OR FIELD
 546 003216  001012                          BNE     3$                  ;FIELD COMPARE AGAINST 77750
 547 003220  026700  000350                  CMP     FSTBSC,R0           ;MANUFACTURING, AT END
 548 003224  001343                          BNE     2$                  ;NO, GO BACK AND READ NEXT
 549
 550 003226                          99$:
     003226  104403  000000' 005450'         MSGN$,BEGIN,NOSEC           ;ASCII MESSAGE CALL WITH COMMON HEADER
 551 003234  004567  176332                  JSR     R5,DROP
 552 003240  000167  175200                  JMP     LOOPL
 553
 554 003244  026700  000326          3$:     CMP     LSTBSC,R0           ;AT END OF FIELD BAD
 555 003250  001331                          BNE     2$                  ;NO, GO BACK
 556 003252  000765                          BR      99$                 ;YES GO DROP DRIVE
 557
 558 003254  016701  174644          4$:     MOV     #BUFVA,R1           ;GET WHERE WE READ
 559 003260  062701  000010                  ADD     #10,R1              ;SKIP PAST I.D. ETC.....
 560 003264  012702  000176                  MOV     #126.,R2            ;ONLY 126 ENTRIES
 561 003270  012103                  44$:    MOV     (R1)+,R3            ;GET CYLINDER
 562 003272  100440                          BMI     88$                 ;MINUS WE'RE DONE
 563 003274  012104                          MOV     (R1)+,R4            ;GET TRACK AND SECTOR
 564 003276  000303                          SWAB    R3                  ;ALIGN PROPERLY
 565 003300  106003                          RORB    R3                                                          ;PA005
 566                                 ;PA005  ASR     R3                                                          ;PA005
 567 003302  006003                          ROR     R3                                                          ;PA005
 568 003304  150403                          BISB    R4,R3
 569 003306  032704  000400                  BIT     #400,R4
 570 003312  001402                          BEQ     5$
 571 003314  052703  000100                  BIS     #100,R3
 572 003320  022767  005373' 002134  5$:     CMP     #HCRC,HTYPE         ;IS ERROR HCRC?
 573 003326  001012                          BNE     6$                  ;NO, GO LOOK FOR BAD SECTOR
 574 003330  022767  000110  000262          CMP     #110,FUNC           ;WE'RE WE DOING READ HEADER
```

```
003336  001006                          BNE     6$              ;NO, GO LOOK FOR BAD SECTOR
003340  042703  000077                  BIC     #77,R3          ;YES, CLEAR SCETOR BITS
003344  020367  000252                  CMP     R3,HDRWD        ;BAD SECTOR
003350  001404                          BEQ     7$
003352  000406                          BR      8$
003354  020367  000174          6$:     CMP     R3,T.DA         ;IS THIS ONE IT???????
003360  001003                          BNE     8$              ;NO
003362  005267  000202          7$:     INC     FND
003366  000412                          BR      9$
003370  005302          8$:             DEC     R2              ;CHECKED WHOLE FILE
003372  001336                          BNE     44$             ;NO
003374  005767  000164          88$:    TST     MFLG            ;WHICH WE DOING
003400  001005                          BNE     9$              ;FIELD , WE'RE DONE
003402  005267  000156                  INC     MFLG            ;MANUFACT. THEN SET UP FIELD
003406  016700  000162                  MOV     FSTBSC,R0
003412  000650                          BR      2$

003414  016700  000202          9$:     MOV     HDRWD,R0
003420  016701  000156                  MOV     LSTCYL,R1
003424  042700  000100                  BIC     #100,R0
003430  160001                          SUB     R0,R1
003432  010177  000100                  MOV     R1,@RLDA
003436  052777  000001  000072          BIS     #1,@RLDA
003444  032767  000100  000150          BIT     #100,HDRWD
003452  001403                          BEQ     10$
003454  052777  000020  000054          BIS     #20,@RLDA
003462  016767  000140  000074  10$:    MOV     DRIVE,MFLG
003470  052767  000006  000066          BIS     #6,MFLG
003476  016777  000062  000026          MOV     MFLG,@RLCS      ; SEEK.
003504  004767  175762                  JSR     PC,WIRDY
003510  005767  000054                  TST     FND
003514  001002                          BNE     11$
003516  000167  177160                  JMP     RPIERR
003522  000167  177270          11$:    JMP     NORPT

                                ;LOCATIONS USED BY MODULE
                                ;
003526  000000          RETRY:  .WORD   0
003530  000003          LIMIT:  .WORD   3
003532  000000          RLCS:   .WORD   0
003534  000000          RLBA:   .WORD   0
003536  000000          RLDA:   .WORD   0
003540  000000          RLMP:   .WORD   0
003542  177777          RLBAE:  .WORD   177777          ; RLV12 BAE                     ;GP-4
003544  177777                  .WORD   177777          ; TERMINATOR.
003546  000000          TMP:    .WORD   0
003550  000000          T.CS:   .WORD   0
003552  000000          T.BA:   .WORD   0
003554  000000          T.DA:   .WORD   0
003556  000000          T.MP:   .WORD   0
003560  000000          T.BAE:  .WORD   0               ; RLV12 BAE                     ;GP-4
003562  000000          T.STAT: .WORD   0
003564  000000          MFLG:   .WORD   0
003566  000000          RWER:   .WORD   0
003570  000000          FND:    .WORD   0
003572  000000          DRVMSK: .WORD   0
003574  000000          FSTBSC: .WORD   0
```



```
003576  000000          LSTBSC: .WORD   0
003600  000000          MASK:   .WORD   0
003602  000000          LSTCYL: .WORD   0
003604  000000          LSTTRK: .WORD   0
003606  000000          MASK2:  .WORD   0
003610  000000          CLK:    .WORD   0
003612  000000          CNT:    .WORD   0
003614  000000          MULDRV: .WORD   0
003616  000000          DLTCNT: .WORD   0               ;NUMBER OF DATA LATE ERRORS
003620  000000          FUNC:   .WORD   0               ;FUNCTION TO BE PERFORMED
003622  000000          HDRWD:  .WORD   0               ;HEADER WORD (NUMBR, R/W)
003624  000000          DIFWD:  .WORD   0               ;DIFFERENCE WORD (SEEK)
003626  000000          DRIVE:  .WORD   0               ;DRIVE UNDER TEST (BITS 8,9)
003630  000000          DVICE:  .WORD   0               ;WORKING "DVID1"
003632  000000          WCNT1:  .WORD   0               ;WORD COUNT (WRITE)
003634  000000          WCNT2:  .WORD   0               ;WORD COUNT (READ)
003636                  BUFIN:  .BLKW   256.
004636                  BSECPR: .BLKW   25.             ;BAD SECTOR LIST
004720  000000          CURMSG: .WORD   0
004722  000000          CURADR: .WORD   0
004724  000000          NXTADR: .WORD   0
004726  000000          NUMB:   .WORD   0
004730  0054U4'         FNCLST: NULLX                   ; OPCODE 0 UNUSED.              ;GP-2
004732  005073'                 MES8A                   ; 1 = WRT CHK.                  ;GP-2
004734  005107'                 MES9                    ; 2 = RESET OR GET STATUS.
004736  005053'                 MES6                    ; 3 = SEEK.
004740  005037'                 MES5                    ; 4 = READ HEADER.
004742  005065'                 MES8                    ; 5 = WRITE.
004744  005060'                 MES7                    ; 6 = READ.
004746  005404'                 NULLX                   ; 7 = READ NO-HEADER (UNUSED).  ;GP-2

004750  116     117     040     MES1:   .ASCIZ  'NO DRIVES PRESENT %'
004753  104     122     111
004756  126     105     123
004761  040     120     122
004764  105     123     105
004767  116     124     040
004772  045     000
004774  116     117     040     MES2:   .ASCIZ  'NO DRIVES LEFT %'
004777  104     122     111
005002  126     105     123
005005  040     114     105
005010  106     124     040
005013  045     000
005015  104     122     111     MES3:   .ASCIZ  'DRIVE '
005020  126     105     040
005023  000
005024  040     104     122     MES4:   .ASCIZ  ' DROPPED %'
005027  117     120     120
005032  105     104     040
005035  045     000
005037  122     105     101     MES5:   .ASCIZ  'READ HEADER'
005042  104     040     110
005045  105     101     104
005050  105     122     000
005053  123     105     105     MES6:   .ASCIZ  'SEEK'
005056  113     000
```

```
669 005060      122     105     101  MES7:   .ASCIZ  'READ'
    005063      104     000
670 005065      127     122     111  MES8:   .ASCIZ  'WRITE'
    005070      124     105     000
671 005073      127     122     111  MES8A:  .ASCIZ  'WRITE-CHECK'                                  ;GP-2
    005076      124     105     055
    005101      103     110     105
    005104      103     113     000
672 005107      104     122     111  MES9:   .ASCIZ  'DRIVE RESET'
    005112      126     105     040
    005115      122     105     124
    005120      105     124     000
673 005123      040     122     105  MES10:  .ASCIZ  ' RETRY LIMIT EXCEEDED%'
    005126      124     122     131
    005131      040     114     111
    005134      115     111     124
    005137      040     105     130
    005142      103     105     105
    005145      104     105     104
    005150      045     000
674 005152      045     123     105  MES11:  .ASCIZ  '%SEEK TO WRONG CYLINDER%'
    005155      105     113     040
    005160      124     117     040
    005163      127     122     117
    005166      116     107     040
    005171      103     131     114
    005174      111     116     104
    005177      105     122     045
    005202      000
675 005203      104     101     124  MES12:  .ASCIZ  'DATA LATE%'
    005206      101     040     114
    005211      101     124     105
    005214      045     000
676 005216      104     122     111  MES13:  .ASCIZ  'DRIVE ERROR%'
    005221      126     105     040
    005224      105     122     122
    005227      117     122     045
    005232      000
677 005233      123     117     106  MES14:  .ASCIZ  'SOFT ERROR   '
    005236      124     040     105
    005241      122     122     117
    005244      122     040     040
    005247      040     000
678 005251      122     114     130  MES15:  .ASCIZ  'RLX LOAD UNIT DROPPED'
    005254      040     114     117
    005257      101     104     040
    005262      125     116     111
    005265      124     040     104
    005270      122     117     120
    005273      120     105     104
    005276      000
679 005277      110     101     122  MES16:  .ASCIZ  'HARD ERROR   '
    005302      104     040     105
    005305      122     122     117
    005310      122     040     040
    005313      040     000
680 005315      103     101     116  NBDSC:  .ASCIZ  'CANT RECOVER BAD SECTOR FILE%'
```



```
    005320      124     040     122
    005323      105     103     117
    005326      126     105     122
    005331      040     102     101
    005334      104     040     123
    005337      105     103     124
    005342      117     122     040
    005345      106     111     114
    005350      105     045     000
681 005353      117     120     111  UP1:    .ASCIZ  'UP1'
    005356      000
682 005357      104     114     124  DLT:    .ASCIZ  'DLT'
    005362      000
683 005363      104     103     113  DCK:    .ASCIZ  'DCK'
    005366      000
684 005367      110     116     106  HNF:    .ASCIZ  'HNF'
    005372      000
685 005373      110     103     122  HCRC:   .ASCIZ  'HCRC'
    005376      103     000
686 005400      116     130     115  NXM:    .ASCIZ  'NXM'
    005403      000
687 005404      040     000          NULLX:  .ASCIZ  ' '
688 005406      045     000          CR:     .ASCIZ  '%'
689                                          .EVEN
690
691 005410  005015'                  DROPMS: MES3
692 005412  004726'                          NUMR
693 005414  005024'                          MES4
694 005416  177777                          177777
695
696 005420  004774'                  NOLEFT: MES2
697 005422  177777                          177777
698
699 005424  004750'                  ABORT:  MES1
700 005426  177777                          177777
701
702 005430  000000                  EXCEED: .WORD   0
703 005432  005123'                          MES10
704 005434  005406'                          CR
705 005436  177777                          177777
706
707 005440  005277'                  HARD:   MES16
708 005442  000000                  TER1:   .WORD   0
709 005444  005406'                          CR
710 005446  177777                          177777
711
712 005450  005315'                  NOSEC:  NBDSC
713 005452  177777                          177777
714
715 005454  005216'                  DRVERR: MES13
716 005456  177777                          177777
717
718 005460  005233'                  SOFT:   MES14
719 005462  000000                  HTYPE:  .WORD   0
720 005464  005406'                          CR
721 005466  177777                          177777
722
```

```
    005470  005152'          BDSEEK: MES11
    005472  177777                   177777

    005474  005251'          DROPLD: MES15
    005476  177777                   177777
                             ;
                             ;REGISTERS OF RL11
                             ;
    005500  003550'          TABLE:  .WORD   T.CS        ;CONTROL AND STATUS REGISTER
    005502  003552'                  .WORD   T.BA        ;BUS ADDRESS REGISTER
    005504  003554'                  .WORD   T.DA        ;DISK ADDRESS REGISTER
    005506  003556'                  .WORD   T.MP        ;DISK DATA BUFFER ADDRESS
    005510  003562'          TABLEY: .WORD   T.STAT      ;T.BAE  ;HAS STATUS ON DRIVE ERROR      ;GP-4
    005512  177777           TABLEZ: .WORD   177777      ;T.STAT ;TERMINATOR                     ;GP-4
    005514  177777                   .WORD   177777      ;TERMINATOR

    005516  000030           PATCH:  .REPT   24.         ;24 DECIMAL PATCH LOCATIONS             ;PA005
                                     .NLIST
                                     .WORD   0
                                     .LIST
                                     .ENDR
            000001                   .END
```


SYMBOL TABLE

```
ABORT    005424R    DIFWD    003624R    LOOPL    000444R    PDP44 =  100000     SETWBA   001730R
ACSR     000102R    DLT      005357R    LSTBSC   003576R    PDP60 =  004000     SOFCNT   000042R
ADDR     000006R    DLTCNT   003616R    LSTCYL   003602R    PDP70 =  010000     SOFER$=  104406
ADDR22=  001000     DRIVE    003626R    LSTTRK   003604R    PIRQ$ =  000004     SOFPAS   000046R
APTPRE=  000200     DROP     001572R    MAP22$=  104416     POPSP =  005726     SOFT     005460R
ASB      000106R    DROPLD   005474R    MASK     003600R    POPSP2=  022626     SPOINT   000032R
ASTAT    000104R    DROPMS   005410R    MASK2    003606R    PRHMS$=  000002     SPSIZ =  000040
AUTO  =  000010     DRVERR   005454R    MES1     004750R    PRTY  =  000000     SR1      000016R
AWAS     000110R    DRVMSK   003572R    MES10    005123R    PRTY0 =  000000     SR2      000020R
BDSEEK   005470R    DRVRTS   002204R    MES11    005152R    PRTY1 =  000040     SR3      000022R
BEGIN    000000R    DVICE    003630R    MES12    005203R    PRTY2 =  000100     SR4      000024R
BIT0  =  000001     DVID1    000014R    MES13    005216R    PRTY3 =  000140     START    000252R
BIT1  =  000002     EA18     002042R    MES14    005233R    PRTY4 =  000200     STAT     000026R
BIT10 =  002000     EA22     002046R    MES15    005251R    PRTY5 =  000240     SVR0     000062R
BIT11 =  004000     ECCMEM=  000100     MES16    005277R    PRTY6 =  000300     SVR1     000064R
BIT12 =  010000     ENDIT$=  104413     MES2     004774R    PRTY7 =  000340     SVR2     000066R
BIT13 =  020000     END$  =  104410     MES3     005015R    PS    =  177776     SVR3     000070R
BIT14 =  040000     ERRTYP   000106R    MES4     005024R    PSW   =  177776     SVR4     000072R
BIT15 =  100000     EXCEED   005430R    MES5     005037R    PUSH  =  005746     SVR5     000074R
BIT2  =  000004     EXEC     002220R    MES6     005053R    PUSH2 =  024646     SVR6     000076R
BIT3  =  000010     EXIT$ =  104400     MES7     005060R    PWRFLG=  000002     SYSCNT   000052R
BIT4  =  000020     FINI     001622R    MES8     005065R    QMON22=  000010     TABLE    005500R
BIT5  =  000040     FNCLST   004730R    MES8A    005073R    RAND$ =  104417     TABLEY   005510R
BIT6  =  000100     FND      003570R    MES9     005107R    RANNUM   000054R    TABLEZ   005512R
BIT7  =  000200     FNDBSC   003042R    MFLG     003564R    RBUFEA   000130R    TAG      000714R
BIT8  =  000400     FREE     000150R    MODNAM   000000R    RBUFPA   000126R    TAG1     001040R
BIT9  =  001000     FSTBSC   003574R    MODSP    000252R    RBUFSZ   000132R    TAG2     001174R
BREAK$=  104407     FUNC     003620R    MSGN$ =  104403     RBUFVA   000124R    TER1     005442R
BR1      000012R    GETPA$=  104415     MSGS$ =  104402     RDHDR    002164R    TMP      003546R
BR2      000013R    GSTAT    002174R    MSG$  =  104401     READ     002110R    TRPDFD=  000023
BSECHF   004636R    GWBUF$=  104414     MULDRV   003614R    RESTRT   000424R    T.BA     003552R
BTOD$ =  104421     HARD     005440R    NBDSC    005315R    RES1     000056R    T.BAE    003560R
BUFIN    003636R    HCRC     005373R    NCPUOP=  000020     RES2     000060R    T.CS     003550R
CAPRES=  000004     HDRWD    003622R    NOAPTY=  000002     RETRY    003526R    T.DA     003554R
CDATA$=  104412     HNF      005367R    NOLEFT   005420R    RH70  =  001000     T.MP     003556R
CDERCT   000144R    HRDCNT   000044R    NORPT    003016R    RLBA     003534R    T.STAT   003562R
CDWDCT   000146R    HRDER$=  104405     NOSEC    005450R    RLBAE    003542R    USTACK=  000001
CHKDRV   000502R    HRDPAS   000050R    NULL  =  000000     RLCS     003532R    VECTOR   000010R
CHKDR1   000506R    HRDRPT   003024R    NULLX    005404R    RLDA     003536R    WASADR   000104R
CKHNG$=  000001     HTYPE    005462R    NUMB     004726R    RLMP     003540R    WBUFEA   000136R
CLK      003610R    ICONT    000036R    NXM      005400R    RPTERR   002702R    WBUFPA   000134R
CLKPRE=  000001     ICOUNT   000040R    NXTADR   004724R    RSTRT    000112R    WBUFRQ   000140R
CLKSP$=  104422     IDNUM    000122R    OPEN  =  000000     RSTRT1   000434R    WBUFSZ   000142R
CNT      003612R    IMODX.=  000000     OPI      005353R    RWER     003566R    WCNT1    003632R
CONFIG   000056R    INDPAR=  000040     OTOA$ =  104420     R6      =%000006    WCNT2    003634R
CR       005406R    INIT     000030R    PARPRE=  002000     R7      =%000007    WDFR     000116R
CSRA     000100R    INTR     000120R    PASCNT   000034R    SHADR    000102R    WDTD     000114R
CURADR   004722R    INTSRV   002262R    PATCH    005516R    SEEK     002140R    WRCHK    002050R
CURMSG   004720R    KTPRES=  000400     PA18     002040R    SETCMN   001762R    WRITE    002060R
DATCK$=  104411     KTXTND=  040000     PA22     002044R    SETRBA   001746R    WTRDY    001472R
DATER$=  104404     LIMIT    003530R    PDPF11=  000002     SETUP    001626R    XFLAG    000005R
DCK      005363R    LOOP     000702R    PDPLSI=  020000

. ABS.   000000     000
         005576     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13181 WORDS  ( 52 PAGES)
```

```
RLAG DEC/X11 SYSTEM EXERCISER M MACRO M1200  02-SEP-82 14:57  PAGE 3-15
SYMBOL TABLE

DYNAMIC MEMORY:  19748 WORDS  ( 75 PAGES)
ELAPSED TIME:  00:00:34
XRLAGO,XRLAGO/CR/-SP=DLXCOM,XRLAGO
```



```
XRLAGO     CREATED BY  MACRO  ON 2-SEP-82 AT 14:57      PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   V01
SYMBOL  VALUE          REFERENCES
ABORT     005424 R      3-183    #3-699
ACSR      000102 R     #3-158    *3-285    *3-434    3-435
ADDR      000006 R     #3-158     3-355
ADDR22  = 001000       #3-158
APIPRE  = 000200       #3-158
ASR       000106 R     #3-158
ASTAT     000104 R     #3-158    *3-286
AUTO    = 000010       #3-158
AWAS      000110 R     #3-158
BOSEEK    005470 R      3-287    #3-723
BEGIN     000000 R     #3-158     3-180     3-183     3-189     3-197     3-235     3-287     3-290     3-300
                        3-312     3-334     3-334     3-338     3-345     3-349     3-351     3-387     3-427
                        3-431     3-455     3-455     3-459     3-461     3-464     3-464     3-492     3-494
                        3-505     3-506     3-510     3-550
BIT0    = 000001       #3-158     3-314
BIT1    = 000002       #3-158     3-232
BIT10   = 002000       #3-158     3-472
BIT11   = 004000       #3-158     3-475     3-483
BIT12   = 010000       #3-158     3-479
BIT13   = 020000       #3-158     3-468
BIT14   = 040000       #3-158     3-449
BIT15   = 100000       #3-158
BIT2    = 000004       #3-158     3-490
BIT3    = 000010       #3-158     3-363     3-385     3-439
BIT4    = 000020       #3-158
BIT5    = 000040       #3-158
BIT6    = 000100       #3-158
BIT7    = 000200       #3-158
BIT8    = 000400       #3-158
BIT9    = 001000       #3-158
BREAKS  = 104407       #3-158     3-334     3-334     3-455     3-455     3-464     3-464
BR1       000012 R     #3-158     3-371
BR2       000013 R     #3-158
BSECHF    004636 R     #3-649
BTUDS   = 104421       #3-158
BUFIN     003636 R      3-158    #3-648
CAPRES  = 000004       #3-158
CDATAS  = 104412       #3-158     3-312
CDERCT    000144 R     #3-158
CDWDCT    000146 R     #3-158
CHKDRV    000502 R     #3-197
CHKDR1    000506 R      3-190     3-193    #3-198
CKHNGS  = 000001       #3-158
CLK       003610 R     #3-330    *3-335    #3-637
CLKPRF  = 000001       #3-158
CLKSPS  = 104422       #3-158
CNT       003612 R      3-185    *3-205    *3-320     3-321    #3-638
CONFIG    000056 R     #3-158
CR        005406 R     #3-688     3-704     3-709     3-720
CSRA      000100 R     #3-158    #3-284    *3-433
CURADR    004722 R     *3-279    *3-280     3-281     3-292    #3-651
CURMSG    004720 R     #3-650
```

XRLAGO CREATED BY MACRO ON 2-SEP-82 AT 14:57 PAGE 2
SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES							
DATCKS	= 104411		#3-158							
DATERS	= 104404		#3-158							
DCK	005363	R	3-486	#3-683						
DIFWD	003624	R	*3-225	*3-239	*3-240	*3-242	*3-244	*3-245	*3-248	*3-257	*3-264
			3-267	3-272	3-412	#3-643					
DLT	005357	R	3-488	#3-682							
DLTCNT	003616	R	*3-166	#3-640							
DRIVE	003626	R	*3-164	*3-196	*3-199	3-329	3-423	3-453	3-524	3-538	3-601
			#3-644								
DROP	001572	R	3-316	3-339	#3-345	3-551					
DROPLD	005474	R	3-180	#3-726							
DROPMS	005410	R	3-345	#3-691							
DRVERM	005354	R	3-459	#3-715							
DRVMSK	003572	R	*3-165	*3-191	3-192	*3-194	3-201	3-346	#3-630		
DRVRTS	002204	R	3-210	#3-419							
DVICE	003630	R	*3-168	3-177	*3-179	3-181	3-201	*3-346	#3-645		
DVI01	000014	R	#3-158	3-168							
EAIR	002042	R	*3-378	*3-381	3-384	*3-393					
EA22	002046	R	3-389	#3-395							
ECCMEM	= 000100		#3-158								
ENDITS	= 104413		#3-158	3-197							
END$	= 104410		#3-158	3-351							
ERRTYP	000106	R	#3-159	*3-289	*3-337	*3-460	*3-493				
EXCEED	005430	R	*3-503	3-506	#3-702						
EXEC	002220	R	3-405	3-410	3-414	3-416	#3-422				
EXITS	= 104400		#3-158	3-427							
FINI	001622	R	3-184	#3-351							
FNCLST	004730	R	3-503	#3-654							
FND	003570	R	*3-533	*3-582	3-605	#3-629					
FNDBSC	003042	R	3-478	3-482	3-487	#3-518					
FREE	000150	R	#3-158								
FSTBSC	003574	R	*3-212	*3-224	3-547	3-589	#3-631				
FUNC	003620	R	*3-399	*3-401	*3-406	*3-411	*3-415	*3-420	3-424	3-501	3-574
			#3-641								
GETPAS	= 104415		#3-158	3-189							
GSTAT	002174	R	#3-417								
GWBUFS	= 104414		#3-158	3-300							
HARD	005440	R	3-505	3-510	#3-707						
HCRC	005373	R	3-477	3-572	#3-685						
HDRRD	003622	R	*3-227	*3-234	3-240	*3-249	*3-250	*3-253	3-254	*3-256	3-260
			*3-262	*3-263	*3-269	*3-271	*3-274	3-281	*3-292	3-294	3-403
			3-408	3-519	3-577	3-592	3-598	#3-642			
HNF	005367	R	3-481	#3-684							
HRDCNT	000044	R	#3-158								
HRDERS	= 104405		#3-158	3-290	3-338	3-461					
HRDPAS	000050	R	#3-158								
HRDRPT	003024	R	3-471	3-480	#3-509						
HTYPE	005462	R	*3-468	*3-470	*3-474	*3-477	*3-481	*3-486	*3-488	3-504	3-509
			3-572	#3-719							
ICUNT	000036	R	#3-158								
ICUUNT	000040	R	#3-158								
IDNUM	000122	R	#3-158								

XRLAGO CREATED BY MACRO ON 2-SEP-82 AT 14:57 PAGE 3
SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES							
IMODX.	= 000000		#3-158	3-300						
INDPAR	= 000040		#3-158							
INIT	000030	R	#3-158							
INTR	000120	R	#3-158	*3-162						
INTSRV	002262	R	3-425	#3-431						
KIPRES	= 000400		#3-158							
KTXTND	= 040000		#3-158							
LIMIT	003530	R	3-495	#3-613						
LOOP	000702	R	#3-226	3-323						
LOOPL	001444	R	#3-191	3-202	3-317	3-324	3-341	3-552		
LSTBSC	003576	R	*3-213	*3-222	3-554	#3-632				
LSTCYL	003602	R	*3-215	*3-220	3-260	3-518	3-593	#3-634		
LSTTRK	003604	R	*3-216	*3-221	3-294	3-531	#3-635			
MAP22$	= 104416		#3-158	3-387						
MASK	003600	R	*3-214	*3-219	3-238	#3-633				
MASK2	003606	R	*3-211	*3-223	3-250	#3-636				
MES1	004750	R	#3-668	3-699						
MES10	005123	R	#3-673	3-703						
MES11	005152	R	#3-674	3-723						
MES12	005203	R	#3-675							
MES13	005216	R	#3-676	3-715						
MES14	005233	R	#3-677	3-718						
MES15	005251	R	#3-678	3-726						
MES16	005277	R	#3-679	3-707						
MES2	004774	R	#3-664	3-696						
MES3	005015	R	#3-665	3-691						
MES4	005024	R	#3-666	3-693						
MES5	005037	R	3-658	#3-667						
MES6	005053	R	3-657	#3-668						
MES7	005060	R	3-660	#3-669						
MES8	005065	R	3-659	#3-670						
MES8A	005073	R	3-655	#3-671						
MES9	005107	R	3-656	#3-672						
MFLG	003564	R	*3-524	*3-525	3-526	*3-532	3-545	3-586	*3-588	*3-601	*3-602
			3-603	#3-627							
MODNAM	000000	R	#3-158								
MODSP	000252	R	3-158	#3-158							
MSG%$	= 104403		#3-158	3-180	3-183	3-287	3-345	3-349	3-459	3-492	3-505
			3-506	3-510	3-550						
MSGSS	= 104402		#3-158								
MSG$	= 104401		#3-158								
MULDRV	003614	R	#3-639								
NBDSC	005315	R	#3-680	3-712							
NCPUUP	= 000020		#3-158								
NDAPTY	= 000002		#3-158								
NOLERT	005420	R	3-349	#3-696							
NORPT	003016	R	3-467	#3-507	3-511	3-608					
NOSEC	005450	R	3-550	#3-712							
NULL	= 000000		#3-158								
NULLX	005404	R	3-448	3-654	3-661	#3-687					
NUMR	004726	R	*3-163	*3-195	*3-200	*3-653	3-692				
NXM	005400	R	3-470	#3-686							

```
XRLAGO      CREATED BY  MACRO  ON 2-SEP-82 AT 14:57        PAGE 4
SYMBOL CROSS REFERENCE                                     CREF   V01
SYMBOL  VALUE              REFERENCES
NXTADR    004724 R         #3-652
OPEN    = 000000            3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158
                            3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158
                            3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158    3-158
                            3-158   #3-158
OPI       005353 R          3-474   #3-681
OTOA$   = 104420           #3-158
PARPRE  = 002000           #3-158
PASCNT    000034 R         #3-158
PATCH     005516 R         #3-739
PA18      002040 R         *3-377   *3-380    3-382    3-387   #3-392
PA22      002044 R          3-388   #3-394
PDPF11  = 000002           #3-158
PDPLSI  = 020000           #3-158
PDP44   = 100000           #3-158
PDP60   = 004000           #3-158
PDP70   = 010000           #3-158
PIRQ$   = 000004           #3-158    3-431
POPSP   = 005726           #3-158
POPSP2  = 022626           #3-158
PRHMS$  = 000002           #3-158
PRTY    = 000000           #3-158
PRTY0   = 000000            3-158   #3-158
PRTY1   = 000040           #3-158
PRTY2   = 000100           #3-158
PRTY3   = 000140           #3-158
PRTY4   = 000200           #3-158
PRTY5   = 000240            3-158   #3-158
PRTY6   = 000300           #3-158
PRTY7   = 000340           #3-158
PS      = 177776           #3-158
PSW     = 177776           #3-158
PUSH    = 005746           #3-158
PUSH2   = 024646           #3-158
PWRFLG  = 000002           #3-158
QMON22  = 000010           #3-158
RAND$   = 104417           #3-158    3-235
RANNUM    000054 R         #3-158    3-236
RBUFEA    000130 R         #3-158    3-381
RBUFPA    000126 R         #3-158    3-312    3-380
RBUFSZ    000132 R         #3-158    3-298
RBUFVA    000124 R         #3-158    3-189    3-558
RDHDR     002164 R          3-226    3-277   #3-415
READ      002110 R          3-309   #3-406
RESTPT    000424 R          3-158   #3-185
RES1      000056 R         #3-158
RES2      000060 R         #3-158
RETRY     003526 R         *3-203   *3-444    3-495   *3-497   *3-507   #3-612
RH70    = 001000           #3-158
RLBA      003534 R         *3-358    3-382    3-388    3-436   #3-615
RLBAE     003542 R         *3-366    3-389    3-441   #3-618
RLCS      003532 R          3-284    3-285    3-286    3-328    3-329    3-331    3-338   *3-356    3-383
```



```
XRLAGO      CREATED BY  MACRO  ON 2-SEP-82 AT 14:57        PAGE 5
SYMBOL CROSS REFERENCE                                     CREF   V01
SYMBOL  VALUE              REFERENCES
                            3-384    3-422    3-423    3-424    3-426    3-433    3-434    3-454    3-456
                            3-463    3-465    3-526    3-536    3-538    3-539    3-541    3-603   #3-614
RLDA      003536 R         *3-360    3-403    3-408    3-412    3-413    3-417    3-419    3-437    3-451
                            3-462    3-522    3-523    3-534    3-596    3-597    3-600   #3-616
RLMP      003540 R         *3-362    3-402    3-407    3-438    3-458    3-535   #3-617
RPTERR    002702 R         #3-490    3-607
RSTRT     000112 R         #3-158
RSTRT1    000434 R          3-182    3-186   #3-188
RWER      003566 R         *3-204    3-304    3-307    3-310   *3-319   *3-447   #3-628
R6      =%000006           #3-158
R7      =%000007           #3-158
SBADR     000102 R         #3-158
SEEK      002140 R          3-275   #3-411
SETCMN    001762 R          3-379   #3-382
SETRBA    001746 R         #3-380    3-409    3-537
SETUP     001626 R          3-167   #3-355
SETWBA    001730 R         #3-377    3-404
SOFCNT    000042 R         #3-158
SOFERS  = 104406           #3-158    3-494
SOFPAS    000046 R         #3-158
SOFI      005460 R          3-492   #3-718
SPOINT    000032 R         #3-158
SPSIZ   = 000040           #1-28     3-158
SR1       000016 R         #3-158    3-232    3-314    3-363    3-385    3-439    3-490
SR2       000020 R         #3-158
SR3       000022 R         #3-158
SR4       000024 R         #3-158
START     000252 R          3-158   #3-160    3-187    3-370
STAT      000026 R         #3-158
SVR0      000062 R         #3-158
SVR1      000064 R         #3-158
SVR2      000066 R         #3-158
SVR3      000070 R         #3-158
SVR4      000072 R         #3-158
SVR5      000074 R         #3-158
SVR6      000076 R         #3-158
SYSCNT    000052 R         #3-158
TABLE     005500 R          3-290    3-461    3-494   #3-731
TABLEY    005510 R         *3-367   #3-735
TABLEZ    005512 R         *3-368   #3-736
TAG       000714 R         #3-232    3-296
TAG1      001040 R          3-233   #3-253
TAG2      001174 R          3-251    3-259    3-266    3-273   #3-275
TER1      005442 R         *3-504   *3-509   #3-708
TMP       003546 R         *3-452   *3-453    3-454    3-463   #3-620
TRPDFD  = 000023           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158    3-158    3-158   #3-158    3-158    3-158
                           #3-158    3-158    3-158   #3-158
```

XRLAGO CREATED BY MACRO ON 2-SEP-82 AT 14:57 PAGE 6
SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES								
T.BA	003552	R	*3-436	#3-622	3-732						
T.MAE	003560	R	3-367	*3-441	#3-625						
T.CS	003550	R	*3-435	3-442	3-449	3-468	3-472	3-475	3-479	3-483	#3-621
			3-731								
T.DA	003554	R	*3-437	*3-485	3-580	#3-623	3-733				
T.MP	003556	R	3-217	3-227	3-279	*3-438	#3-624	3-734			
T.STAT	003562	R	3-368	*3-432	*3-458	#3-626	3-735				
USTACK	= 000001		#3-158								
VECTOR	000010	R	*3-158	3-369	3-425						
WASADR	000104	R	*3-158								
WBUFEA	000136	R	*3-158	3-378							
WBUFPA	000134	R	*3-158	3-377							
WBUFWD	000140	R	#3-158								
WBUFSZ	000142	R	#3-158	3-301							
WCNT1	003632	R	*3-301	*3-302	3-402	#3-646					
WCNT2	003634	R	*3-298	*3-299	3-407	#3-647					
WDER	000116	R	*3-158	*3-160							
WDTO	000114	R	*3-158	*3-161							
WHCHK	002050	R	3-306	#3-399							
WRITE	002060	R	3-303	#3-401							
WTRDY	001472	R	3-209	3-276	#3-328	3-527	3-540	3-604			
XFLAG	000005	R	#3-158								

XRLAGO CREATED BY MACRO ON 2-SEP-82 AT 14:57 PAGE 7
MACRO CROSS REFERENCE CREF V01

MACRO NAME	REFERENCES								
BKMOD	#1-124								
BREAK	#1-222	3-334	3-455	3-464					
BTOD	#1-246								
CKDATA	#1-282	3-312							
CLKSP	#1-149								
DATACK	#1-291								
DATERR	#1-175								
DFSEVN	#1-314	3-158	3-158	3-158	3-158	3-158	3-158	3-158	3-158
	3-156	3-158	3-158						
DSEVNT	#1-324	3-158	3-158	3-158	3-158	3-158	3-158	3-158	
END	#1-212	3-351							
ENDIT	#1-203	3-197							
ENDMOD	#1-208								
EQUATS	#1-330	3-158							
EXIT	#1-157	3-427							
GETPA	#1-273	3-189							
GWBUFF	#1-261	3-300							
HLFBRK	#1-227								
HRDER	#1-165	3-290	3-338	3-461					
IOMOD	#1-120								
IOMODP	#1-144								
IOMODR	#1-140								
IOMODX	#1-136	3-158							
MAP22	#1-277	3-387							
MODULE	#1-29	3-158							
MSG	#1-191								
MSGN	#1-195	3-180	3-183	3-287	3-345	3-349	3-459	3-492	3-505	3-506
	3-510	3-550							
MSGS	#1-199								
NBKMOD	#1-132								
OTOA	#1-232								
PIRQ	#1-216	3-431							
RAND	#1-161	3-235							
SBKMOD	#1-128								

.REM _

IDENTIFICATION

PRODUCT CODE: AC-8962B-MC

PRODUCT NAME: CXPCSB0 PCS-11 MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT

PCS IS AN IOMOD THAT EXERCISES PCS CONTROLLER AND FILE BOX. IT WILL EXERCISE ALL BITS OF CSR ADDRESS, CHECK FOR MAINTENANCE INTERRUPT CAPABILITIES AND CHECK ALL THE ADDRESSES FROM 171000 TO 171375 IN MAINTENANCE MODE.

2. REQUIREMENTS

HARDWARE: IOCM CONTROL MODULE WITH FILE BOX
STORAGE:: PCS REQUIRES:
1. DECIMAL WORDS: 748
2. OCTAL WORDS: 1354
3. OCTAL BYTES: 2730

3. TEST DESCRIPTION

ONE PASS OF THE MODULE CONSISTS OF CSR CHECKS, INTERRUPT TEST AND IAR TEST IN MAINTENANCE MODE. DURING THE TEST DBIT IS SET SO THE TEST WILL NOT EFFECT I/O MODULES

4. EXECUTION TIME

ONE PASS OF THE TEST TAKES LESS THEN 1 SEC

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS

DEUADR: 171376
VECTOR: 234
DEVCNT: 1

6. MODULE OPERATION

TEST SEQUENCE

A. SET UP THE DEVICE ADDRESSSES
B. SET G BIT, READ IT BACK AND CLEAR IT
C. SET D BIT, READ IT BACK AND CLEAR IT
D. SET T BIT, READ IT BACK AND CLEAR IT
E. SET RIE BIT, READ IT BACK AND CLEAR IT
F. SET E BIT, READ IT BACK AND CLEAR IT
G. SET M BIT, READ IT BACK AND CLEAR IT
H. SET MBIT AND READ ALL ADDRESSES FROM
I. SET MAINTENCE INTERRUPT AND READ ALL ONES FROM IAR 171000 TO 171375
J. RESET CSR.

7. OPERATOR OPTIONS:

NONE

8. PRINTOUTS:

```
                        ALL PRINTOUTS ARE STANDARD
                        -
```

```
                                        IOMOD   <PCSB  >,171376,234,0,0,0,10,147
                                        MODULE  140000,PCSB  ,171376,234,0,0,0,10,147
                                        .TITLE  PCSB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       DDXCOM  VERSION 6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************
000000'                         BEGIN:
000000' 041520  041123      040 MODNAM: .ASCII  /PCSB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 171376                  ADDR:   171376+0                ;1ST DEVICE ADDR.
000010' 000234                  VECTOR: 234+0                   ;1ST DEVICE VECTOR.
000012'     500                 BR1:    .BYTE   PRTY0+0         ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************
000326' 140000                  STAT:   140000                  ;STATUS WORD.
000030' 000246'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000010                  ICONT:  10                      ;# OF ITERATIONS PER PASS=10
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000274'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000147                  IDNUM:  147                     ;MODULE IDENTIFICATION NUMBER=147
                                        .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
000040
```



```
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:
                                ;*****************************************************************
```

```
                                ;TEMPORARY STORAGE
000224' 000000          TEMP1:  .WORD   0       ;TEMPORARY STORAGE
000226' 000000          TEMP2:  .WORD   0       ;TEMPORARY STORAGE
000230' 000000          TEMP3:  .WORD   0       ;TEMPORARY STORAGE
000232' 000000          CSR:    .WORD   0       ;ADDRESS OF CSR = 171377
000234' 000000          IAR:    .WORD   0       ;ADDRESS OF IAR = 17136
000236' 000000          INTFLG: .WORD   0       ;INTERRUPT OCCURED FLAG
000240' 000000          CNT:    .WORD   0       ;PASS COUNT
000242' 000000          BASE:   .WORD   0       ;ADDRESS 171000
000244' 000000          VECT2:  .WORD   0       ;SECEND ADDR OF VECTOR

000246' 012767  000001  177644  START:  MOV     #1,INTR         ; ONE INTERRUPT/ITERATION
000254' 012767  000010  177632          MOV     #8.,WDTO        ;8. WORDS TO MEM/ITERATION
000262' 012767  000010  177626          MOV     #8.,WDFR        ;8. WORDS FROM MEM/ITERATION
000270' 005067  177744                  CLR     CNT             ;CLEAR PASS COUNT
000274' 016700  177506          RESTRT: MOV     ADDR,R0
000300' 010067  177730                  MOV     R0,IAR          ;SET ADDRESS OF IAR = 171376
000304' 005200                          INC     R0
000306' 010067  177720                  MOV     R0,CSR          ;SET ADDRESS OF CSR = 171377
000312' 042700  000377                  BIC     #377,R0
000316' 010067  177720                  MOV     R0,BASE         ;SET ADDRESS OF THE LOWEST ID = 171000

000322' 012703  000033          1$:     MOV     #9. * 3.,R3     ;SET UP WAIT LOOP
000326' 152777  000002  177676          BISB    #2,@CSR         ;SET CRIT TO CLEAR CSR
000334' 000240          32$:    NOP                             ;WAIT FOR CLEAR
000336' 005303                  DEC     R3
000340' 001375                  BNE     32$
000342' 012703  000044          33$:    MOV     #12.*3.,R3      ;ANOTHER WAIT LOOP
000346' 152777  000002  177656          BISB    #2,@CSR
000354' 000240          34$:    NOP
000356' 005303                  DEC     R3
000360' 001375                  BNE     34$
000362' 105777  177644                  TSTB    @CSR            ;IS CSR CLEAR?
000366' 001407                  BEQ     2$
000370' 117767  177636  177512          MOVB    @CSR,AWAS               ;NO, SET BAD DATA FOR ERROR CALL
000376' 005067  177504                  CLR     ASB             ;SET GOOD DATA.

                                ;*****************************************************************
000402' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************

                        ;TEST   GBIT
000406' 152777  000001  177616  2$:     BISB    #1,@CSR                 ;SET RIF BIT
000414' 132777  000001  177610          BITB    #1,@CSR                 ;IS IT SET
000422' 001017                  BNE     3$                      ;YES, GO TO 3$
000424' 117767  177602  177450          MOVB    @CSR,ACSR               ;LOAD CONTENTS OF CSR FOR ERROR
000432' 016767  177574  177440          MOV     CSR,CSRA                ;LOAD ADDRESS OF CSR FOR ERROR
000440' 104403  000000' 002056'         MSGNS,BEGIN,RIFNOT      ;ASCII MESSAGE CALL WITH COMMON HEADER

000446' 012767  000025  177432          MOV     #25,ERRTYP      ;BIT STUCK
                                ;*****************************************************************
000454' 104405  000000' 000000          HRDER$,BEGIN,NULL       ; RIF BIT IS NOT SETTING
                                ;*****************************************************************

000462' 132777  000001  177542  3$:     BITB    #1,@CSR                 ;NOW RIF BIT SHOULD BE CLEAR
000470' 001417                  BEQ     4$                      ;YES GO TO 4$
```



```
000472' 117767  177534  177402          MOVB    @CSR,ACSR
000500' 016767  177526  177372          MOV     CSR,CSRA
000506' 104403  000000' 002062'         MSGNS,BEGIN,RIFCLR      ;ASCII MESSAGE CALL WITH COMMON HEADER
000514' 012767  000025  177364          MOV     #25,ERRTYP              ;BIT STUCK
                                ;*****************************************************************
000522' 104405  000000' 000000          HRDER$,BEGIN,NULL       ; RIF BIT IS NOT CLEARING
                                ;*****************************************************************

000530' 152777  000004  177474  4$:     BISB    #4,@CSR                 ;SET GBIT AT CSR
000536' 132777  000004  177466          BITB    #4,@CSR                 ;TEST IF SET
000544' 001017                  BNE     5$                      ;YES,GO TO 5$
000546' 117767  177460  177326          MOVB    @CSR,ACSR
000554' 016767  177452  177316          MOV     CSR,CSRA
000562' 104403  000000' 002066'         MSGNS,BEGIN,GBITS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000570' 012767  000025  177310          MOV     #25,ERRTYP              ;BIT STUCK
                                ;*****************************************************************
000576' 104405  000000' 000000          HRDER$,BEGIN,NULL       ; GENERIC BIT IS NOT SETTING
                                ;*****************************************************************

000604' 142777  000004  177420  5$:     BICB    #4,@CSR                 ;CLEAR GBIT AT CSR
000612' 132777  000004  177412          BITB    #4,@CSR                 ;TEST IF CLEAR
000620' 001417                  BEQ     6$                      ;YES, GO TO 6$
000622' 117767  177404  177252          MOVB    @CSR,ACSR
000630' 016767  177376  177242          MOV     CSR,CSRA
000636' 104403  000000' 002072'         MSGNS,BEGIN,GBITC       ;ASCII MESSAGE CALL WITH COMMON HEADER
000644' 012767  000025  177234          MOV     #25,ERRTYP              ;BIT STUCK
                                ;*****************************************************************
000652' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;GENERIC BIT IS NOT CLEARING
                                ;*****************************************************************

000660' 152777  000020  177344  6$:     BISB    #20,@CSR                ;SET DBIT AT CSR
000666' 132777  000020  177336          BITB    #20,@CSR                ;TEST IF SET
000674' 001217                  BNE     7$                      ;YES,GO TO 7$
000676' 117767  177330  177176          MOVB    @CSR,ACSR
000704' 016767  177322  177166          MOV     CSR,CSRA
000712' 104403  000000' 002076'         MSGNS,BEGIN,DBITS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000720' 012767  000025  177160          MOV     #25,ERRTYP              ;BIT STUCK
                                ;*****************************************************************
000726' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;DBIT IS NOT SETTING
                                ;*****************************************************************

000734' 152777  000010  177270  7$:     BISB    #10,@CSR                ;SET TBIT @CSR
000742' 132777  000010  177262          BITB    #10,@CSR                ;IS IT SET
000750' 001017                  BNE     8$                      ;YES, CONT.
000752' 117767  177254  177122          MOVB    @CSR,ACSR
000760' 016767  177246  177112          MOV     CSR,CSRA
000766' 104403  000000' 002102'         MSGNS,BEGIN,TBITS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000774' 012767  000025  177104          MOV     #25,ERRTYP              ;BIT STUCK
                                ;*****************************************************************
001002' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;TBIT NOT SETTING
                                ;*****************************************************************

001010' 142777  000010  177214  8$:     BICB    #10,@CSR                ;CLEAR TBIT
001016' 132777  000010  177206          BITB    #10,@CSR                ;IS IT CLEAR
001024' 001417                  BEQ     9$                      ;YES, CONT
001026' 117767  177200  177046          MOVB    @CSR,ACSR
```

```
001034' 016767  177172  177036'        MOV     CSR,CSRA
001042' 104403  000000' 002106'        MSGNS,BEGIN,TBITC       ;ASCII MESSAGE CALL WITH COMMON HEADER
001050' 012767  000025  177030         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001056' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;TBIT NOT CLEARING
                                       ;*****************************************************************

001064' 012703  000025          9$:    MOV     #7.*3.,R3               ;SET UP WAIT LOOP
001070' 142777  000020  177134         BICB    #20,@CSR                ;CLEAR DBIT
001076' 000240                 50$:    NOP                             ;WAIT FOR CSR
001100' 005303                         DEC     R3
001102' 001375                         BNE     50$
001104' 132777  000020  177120         BITB    #20,@CSR
001112' 001420                         BEQ     10$
001114' 001417                         BEQ     10$
001116' 117767  177110  176756         MOVB    @CSR,ACSR
001124' 016767  177102  176746         MOV     CSR,CSRA
001132' 104403  000000' 002114'        MSGNS,BEGIN,DBITC       ;ASCII MESSAGE CALL WITH COMMON HEADER
001140' 012767  000025  176740         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001146' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;DBIT NOT CLEARING
                                       ;*****************************************************************

001154' 152777  000040  177050  10$:    BISB    #40,@CSR                ;SET MBIT
001162' 132777  000040  177042         BITB    #40,@CSR                ;IS IT SET
001170' 001017                         BNE     11$
001172' 117767  177034  176702         MOVB    @CSR,ACSR               ;NO
001200' 016767  177026  176672         MOV     CSR,CSRA
001206' 104403  000000' 002114'        MSGNS,BEGIN,MBITS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001214' 012767  000025  176664         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001222' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;MAINTENANCE BIT IS NOT SETTING
                                       ;*****************************************************************

001230' 142777  000040  176774  11$:    BICB    #40,@CSR                ;CLEAR MBIT
001236' 132777  000040  176766         BITB    #40,@CSR                ;IS IT CLEAR
001244' 001417                         BEQ     12$
001246' 117767  176760  176626         MOVB    @CSR,ACSR               ;NO
001254' 016767  176752  176616         MOV     CSR,CSRA
001262' 104403  000000' 002120'        MSGNS,BEGIN,MBITC       ;ASCII MESSAGE CALL WITH COMMON HEADER
001270' 012767  000025  176610         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001276' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;MAINTENANCE BIT IS NOT CLEARING
                                       ;*****************************************************************

001304' 152777  000100  176720  12$:    BISB    #100,@CSR               ;SET EBIT
001312' 132777  000100  176712         BITB    #100,@CSR
001320' 001017                         BNE     13$
001322' 117767  176704  176552         MOVB    @CSR,ACSR
001330' 016767  176676  176546         MOV     CSR,CSRA
001336' 104403  000000' 002124'        MSGNS,BEGIN,EBITS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001344' 012767  000025  176534         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001352' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;INTERRUPT ENABLE BIT IS NOT SETTING
                                       ;*****************************************************************
```



```
001360' 142777  000100  176644  13$:    BICB    #100,@CSR               ;CLEAR EBIT
001366' 132777  000100  176636         BITB    #100,@CSR
001374' 001417                         BEQ     14$
001376' 117767  176630  176476         MOVB    @CSR,ACSR
001404' 016767  176622  176466         MOV     CSR,CSRA
001412' 104403  000000' 002130'        MSGNS,BEGIN,EBITC       ;ASCII MESSAGE CALL WITH COMMON HEADER
001420' 012767  000025  176460         MOV     #25,ERRTYP              ;BIT STUCK
                                       ;*****************************************************************
001426' 104405  000000' 000000         HRDERS,BEGIN,NULL       ;INTERRUPT ENABLE BIT IS NOT CLEARING
                                       ;*****************************************************************

                                       ;THIS TEST WILL CHECK ALL ADDRESSES WITH MBIT SET

001434' 112777  000040  176570  14$:    MOVB    #40,@CSR                ;SET MBIT
001442' 016702  176574                 MOV     BASE,R2
001446' 005000                         CLR     R0
001450' 005001                         CLR     R1
001452' 152777  000001  176552  15$:    BISB    #1,@CSR                 ;SET RIF BIT
001460' 112201                         MOVB    (R2)+,R1                ;READ ADDRESS 171000+R0 AND
001462' 042701  177400                 BIC     #177400,R1
                                                                       ;LOAD CONTENTS INTO R1

001466' 220001                         CMP     R0,R1                   ;R1 SHOULD BE EQUAL R2
001470' 001416                         BEQ     16$                     ;YES
001472' 010067  176410                 MOV     R0,ASR                  ;NO, SAVE GOOD DATA
001476' 010167  176406                 MOV     R1,AWAS                 ;     SAVE BAD DATA
001502' 016767  176534  176372         MOV     BASE,ACSR               ;     SAVE ADDRESS
001510' 150067  176366                 BISB    R0,ACSR
001514' 016767  176512  176356         MOV     CSR,CSRA
                                       ;*****************************************************************
001522' 104404  000000'                DATERS,BEGIN            ;DATA ERROR!!!
                                       ;*****************************************************************
001526' 005200                 16$:    INC     R0                      ;GO TO NEXT ADDRESS
001530' 122700  000376                 CMPB    #376,R0                 ;IS IT LAST ONE
001534' 001346                         BNE     15$                     ;NO, DO IT AGAIN

                                       ;THIS TEST CHECKS MAINTENANCE INTERRUPT
                                       ;IF MBIT & EBIT ARE SET 10CM GENERATES
                                       ;INTERRUPT AT ADDRESS 234 AND IAR HAS
                                       ;UPPER BYTE OF CSR ADDRESS (377)

001536' 012777  001672' 176244  17$:    MOV     #20$,@VECTOR            ;SET INTERRUPT VECTOR
001544' 016767  176240  176472         MOV     VECTOR,VECT2            ;SET VECTOR + 2
001552' 062767  000002  176464         ADD     #2,VECT2
001560' 116777  176226  176456         MOVB    RR1,@VECT2              ;SET PRIORITY LEVEL ON INTERRUPT
001566' 005067  176444                 CLR     INTFLG                  ;CLEAR INTERRUPT FLAG
001572' 012767  000005  176424         MOV     #5,TEMP1                ;SET COUNT FOR TIMEOUT
001600' 152777  000140  176424         BISB    #140,@CSR               ;ENABLE INTERRUPT

001606'                        18$:
001606' 104407  000000'                BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001612' 104407  000000'                BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.

001616' 005767  176414                 TST     INTFLG                  ;INTERRUPT OCCURR?
001622' 001431                         BNE     21$                     ;YES, 21$
001624' 005367  176374                 DEC     TEMP1                   ;NO, IS IT TIMEOUT
```

```
001630' 001366                          BNE     1B$                     ;NO, LOOP TO BREAK

001632' 117767  176374  176242          MOVB    @CSR,ACSR
001640' 016767  176366  176232          MOV     CSR,CSRA
001646' 104403  000000' 002134'         MSGN$,BEGIN,NOINT       ;ASCII MESSAGE CALL WITH COMMON HEADER
001654' 012767  000023  176224          MOV     #23,ERRTYP              ;NO INTERRUPT
                                        ;*************************************************************
001662' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;NO INTERRUPT
                                        ;*************************************************************

001670' 000444                          BR      22$

001672' 142777  000100  176332  20$:    BICB    #100,@CSR               ;CLEAR INTERRUPT ENABLE
001700' 005267  176332                  INC     INTFLG                  ;GET INTERRUPT FLAG
001704' 000002                          RTI

001706' 152777  000001  176316  21$:    BISB    #1,@CSR                 ;SET RIF BIT
001714' 117700  176314                  MOVB    @IAR,R0                 ;CHECK IF IAR = 377
001720' 122700  000377                  CMPB    #377,R0
001724' 001426                          BEQ     22$                     ;YES, GO TO END
001726' 116067  176156                  MOVB    R0,AWAS                 ;NO, SAVE BAD DATA
001732' 012767  000377  176146          MOV     #377,ASR                ;    SAVE GOOD DATA
001740' 117767  176270  176134          MOVB    @IAR,ACSR               ;    SAVE ADDRESS
001746' 016767  176262  176124          MOV     IAR,CSRA
001754' 104403  000000' 002140'         MSGN$,BEGIN,IARERR      ;ASCII MESSAGE CALL WITH COMMON HEADER
001762' 012767  000001  176116          MOV     #1,ERRTYP               ;DATA ERROR
                                        ;*************************************************************
001770' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;WRONG DATA IN IAR AFTER INTERRUPT
                                        ;*************************************************************
                                        ;*************************************************************
001776' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*************************************************************

002002' 012703  000041          22$:    MOV     #11.*3.,R3              ;SET UP WAIT LOOP
002006' 152777  000002  176216          BISB    #2,@CSR                 ;CLEAR CSR
002014' 000240          60$:    NOP                             ;WAIT
002016' 005303                  DEC     R3
002020' 001375                  BNE     60$                     ;WAIT SOME MORE
002022' 012703  000036          MOV     #10.*3.,R3              ;SET UP ANOTHER LOOP
002026' 152777  000002  176176  BISB    #2,@CSR                 ;CLEAR AGAIN
002034' 000240          65$:    NOP
002036' 005303                  DEC     R3
002040' 001375                  BNE     65$
002042'                 23$:
002042' 104413  000000'         ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
002046' 000167  176222          JMP     RESTRT
002052'                 FINI:
002052' 104410  000000'         END$,BEGIN              ;

                                .EVEN

002056' 002144'                 RIFNOT: MES1
002060' 177777                          177777
```



```
002062' 002173'                 RIFCLR: MES2
002064' 177777                          177777

002066' 002223'                 GBITS:  MES3
002070' 177777                          177777

002072' 002263'                 GBITC:  MES4
002074' 177777                          177777

002076' 002324'                 DBITS:  MES5
002100' 177777                          177777

002102' 002350'                 TBITS:  MES6
002104' 177777                          177777

002106' 002374'                 TBITC:  MES7
002110' 177777                          177777

002112' 002421'                 DBITC:  MES8

002114' 002446'                 MBITS:  MES9
002116' 177777                          177777

002120' 002472'                 MBITC:  MES10
002122' 177777                          177777

002124' 002517'                 EBITS:  MES11
002126' 177777                          177777

002130' 002563'                 EBITC:  MES12
002132' 177777                          177777

002134' 002630'                 NOINT:  MES13
002136' 177777                          177777

002140' 002666'                 IARERR: MES14
002142' 177777                          177777

002144' 044522  020106  044502  MES1:   .ASCIZ  "RIF BIT IS NOT SETTING"
002152' 020124  051511  047040
002160' 052117  051440  052105
002166' 044524  043516     000

002173'    122  043111  041040  MES2:   .ASCIZ  "RIF BIT IS NOT CLEARING"
002200' 052111  044440  025123
002206' 047516  020124  046103
002214' 040505  044522  043516
002222'     000

002223'    107  047105  051105  MES3:   .ASCIZ  "GENERIC CODE BIT IS NOT SETTING"
002230' 041511  041440  042117
002236' 020105  044502  020124
002244' 051511  047040  052117
002252' 051440  052105  044524
002260' 043516     000

002263'    107  047105  051105  MES4:   .ASCIZ  "GENERIC CODE BIT IS NOT CLEARING"
002270' 041511  041440  042117
002276' 020105  044502  020124
```

```
002304' 051511  047040  052117
002312' 041440  042514  051101
002320' 047111  000107
002324' 041104  052111  044440   MES5:   .ASCIZ  "DBIT IS NOT SETTING"
002332' 020123  047516  020124
002340' 042523  052124  047111
002346' 000107
002350' 041124  052111  044440   MES6:   .ASCIZ  "TBIT IS NOT SETTING"
002356' 020123  047516  020124
002364' 042523  052124  047111
002372' 000107
002374' 041124  052111  044440   MES7:   .ASCIZ  "TBIT IS NOT CLEARING"
002402' 020123  047516  020124
002410' 046103  040505  044522
002416' 043516     000
002421'    104  044502  020124   MES8:   .ASCIZ  "DBIT IS NOT CLEARING"
002426' 051511  047040  052117
002434' 041440  042514  051101
002442' 047111  000107
002446' 041115  052111  044440   MES9:   .ASCIZ  "MBIT IS NOT SETTING"
002454' 020123  047516  020124
002462' 042523  052124  047111
002470' 000107
002472' 041115  052111  044440   MES10:  .ASCIZ  "MBIT IS NOT CLEARING"
002500' 020123  047516  020124
002506' 046103  040505  044522
002514' 043516     000
002517'    111  052116  051105   MES11:  .ASCIZ  "INTERRUPT ENABLE BIT IS NOT SETTING"
002524' 052522  052120  042440
002532' 040516  046102  020105
002540' 044502  020124  051511
002546' 047040  052117  051440
002554' 052105  044524  043516
002562'    000
002563'    111  052116  051105   MES12:  .ASCIZ  "INTERRUPT ENABLE BIT IS NOT CLEARING"
002570' 052522  052120  042440
002576' 040516  046102  020105
002604' 044502  020124  051511
002612' 047040  052117  041440
002620' 042514  051101  047111
002626' 000107
002630' 047516  046440  044501   MES13:  .ASCIZ  "NO MAINTENANCE MODE INTERRUPT"
002636' 052116  047105  047101
002644' 042503  046440  042117
002652' 020105  047111  042524
002660' 051122  050125  000124
002666' 051127  047117  020107   MES14:  .ASCIZ  "WRONG DATA IN IAR AFTER INTERRUPT"
002674' 040504  040524  044440
002702' 020116  040511  020122
002710' 043101  042524  020122
002716' 047111  042524  051122
002724' 050125  000124

                                         .EVEN

        000001                           .END
```



```
ACSR      000102R    148#   212*   223*   234*   245*   256*   267*   278*   294*   305*   316*   327*   338*
                     361*   362*   393*   413*
ADDR      000006R    114#   182
ADDR22= 001000       166#
ASB       000106R    152#   202*   359*   412*
ASTAT     000104R    150#
AWAS      000110R    153#   201*   360*   411*
BASE      000242R    175#   187*   349    361
BEGIN     000000R    111#   205    214    218    225    228    236    239    247    250    258    261    269
                     272    280    283    296    299    307    310    318    321    329    332    340    343
                     365    385    386    395    398    415    418    421    435    439
BIT0    = 000001     166#
BIT1    = 000002     166#
BIT10   = 002000     166#
BIT11   = 004000     166#
BIT12   = 010000     166#
BIT13   = 020000     166#
BIT14   = 040000     166#
BIT15   = 100000     166#
BIT2    = 000004     166#
BIT3    = 000010     166#
BIT4    = 000020     166#
BIT5    = 000040     166#
BIT6    = 000100     166#
BIT7    = 000200     166#
BIT8    = 000400     166#
BIT9    = 001000     166#
BREAK$= 104407       166#   385    386
BR1       000012R    116#   379
BR2       000013R    117#
BTOD$   = 104421     166#
CDATA$= 104412       166#
CNT       000240R    174#   181*
CONFIG    000056R    136#
CSR       000232R    171#   185*   190*   195*   199    201    209*   210    212    213    221    223    224
                     231*   232    234    235    242*   243    245    246    253*   254    256    257    264*
                     265    267    268    275*   276    278    279    287*   291    294    295    302*   303
                     305    306    313*   314    316    317    324*   325    327    328    335*   336    338
                     339    348*   352*   363    382*   393    394    403*   407*   425*   430*
CSRA      000100R    146#   213*   224*   235*   246*   257*   268*   279*   295*   306*   317*   328*   339*
                     363*   394*   414*
DATCK$= 104411       166#
DATER$= 104404       166#   205    365    421
DBITC     002112R    296    465#
DBITS     002076R    258    456#
DVID1     000014R    118#
EBITC     002130R    340    476#
EBITS     002124R    329    473#
ENDIT$= 104413       166#   435
END$    = 104410     166#   439
ERRTYP    000106R    151#   216*   226*   237*   248*   259*   270*   281*   297*   308*   319*   330*   341*
                     396*   416*
EXIT$   = 104400     166#
FINI      002052R    438#
GBITC     002072R    247    453#
GBITS     002066R    236    450#
```

```
GETPA$= 104415           166#
GWBUF$= 104414           166#
HRDCNT  000044R          131#
HRDER$= 104405           166#    218     228     239     250     261     272     283     299     310     321     332     343
                                 398     418
HRDPAS  000050R          133#
IAR     000234R          172#    183*    408     413     414
IARERR  002140R          415     482#
ICONT   000036R          128#
ICOUNT  000040R          129#
IDNUM   000122R          158#
INIT    000030R          125#
INTFLG  000236R          173#    380*    388     404*
INTR    000120R          157#    178*
MAP22$= 104416           166#
MBITC   002120R          316     470#
MBITS   002114R          307     467#
MES1    002144R          443     485#
MES10   002472R          470     526#
MES11   002517R          473     530#
MES12   002563R          476     537#
MES13   002630R          479     544#
MES14   002666R          482     549#
MES2    002173R          447     489#
MES3    002223R          450     494#
MES4    002253R          453     500#
MES5    002324R          456     506#
MES6    002350R          459     510#
MES7    002374R          462     514#
MES8    002421R          465     518#
MES9    002446R          467     522#
MODNAM  000000R          112#
MODSP   000224R          126     164#
MSGN$ = 104403           166*    214     225     236     247     258     269     280     296     307     318     329     340
                                 395     415
MSGS$ = 104402           166#
MSG$  = 104401           166#
NOINT   002134R          395     479#
NULL  = 000000           166#    218     228     239     250     261     272     283     299     310     321     332     343
                                 398     418
OPEN  = 000000           113     119     120     121     122     139     140     141     142     143     144     145     146
                                 148     150     152     153     155     156     157     166#
OTOA$ = 104420           166#
PASCNT  000034R          127#
PIRQ$ = 000004           166#
POPSP = 005726           166#
POPSP2= 122626           166#
PRTY  = 000000           166#
PRTY0 = 000000           116     117     166#
PRTY1 = 000040           166#
PRTY2 = 000100           166#
PRTY3 = 000140           166#
PRTY4 = 000200           166#
PRTY5 = 000240           166#
PRTY6 = 000300           166#
PRTY7 = 000340           166#
```



```
PS    = 177776           166#
PSW   = 177776           166#
PUSH  = 005746           166#
PUSH2 = 024546           166#
RAND$ = 104417           166#
RANNUM  000054R          135#
RESTRT  000274R          154     182#    437
RES1    000056R          137#
RES2    000060R          138#
RIFCLR  002062R          225     447#
RIFNOT  002056R          214     443#
RSTRT   000112R          154#
SBADR   000102R          147#
SOFCNT  000042R          130#
SOFER$= 104406           166#
SOFPAS  000046R          132#
SPOINT  000032R          126#
SPSIZ = 000040             1#    159
SR1     000016R          119#
SR2     000020R          120#
SR3     000022R          121#
SR4     000024R          122#
START   000246R          175     178#
STAT    000025R          124#
SVR0    000062R          139#
SVR1    000064R          140#
SVR2    000066R          141#
SVR3    000070R          142#
SVR4    000072R          143#
SVR5    000074R          144#
SVR6    000076R          145#
SYSCNT  000052R          134#
TBITC   002106R          280     462#
TBITS   002102R          269     456#
TEMP1   000224R          168#    381*    390*
TEMP2   000226R          169#
TEMP3   000230R          170#
TRPDFD= 000022           166#
VECTOR  000010R          115#    376*    377
VECT2   000244R          176#    377*    378*    379*
WASADR  000104R          144#
WDFR    000116R          156#    180*
WDTO    000114R          155#    179*
XFLAG   000005R          113#

. ABS.  000000      000
        002730      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XPCSB0,XPCSB0/SOL/CRF:SYM=DDXCOM,XPCSB0
 RUN-TIME: 1 1 .2 SECONDS
 RUN-TIME RATIO: 26/3=6.7
```

CORE USED: 7K (13 PAGES)

```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-E959B-MC

PRODUCT NAME: CXAABR0 AA11-K MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1. ABSTRACT:

"AAR" IS A IOMOD THAT EXERCISES THE AA11-K SCOPE CONTROLLER. A CONFIDENCE LOGIC TEST IS EXECUTED ON THE CONTROL/STATUS, DAC0, DAC1, DAC2 AND DAC3 REGISTERS. ALL LOGIC ERRORS ARE REPORTED TO THE CONSOLE TELETYPE. THE MAJOR PORTION OF THIS MODULE IS DEFERRED TO LEVEL 0 SERVICE. A SIX LETTER MESSAGE (AA11-K) WILL BE PLOTTED ON THE SCREEN DURING EXECUTION. IF A 611/613 SCOPE IS CONNECTED, ALTERNATING CHARACTERS MAY APPEAR TO HAVE ALTERNATING INTENSITY LEVELS. WHEN A VR14 SCOPE IS CONNECTED CHANGING THE CHANNEL SWITCH TO CHANNEL ONE ONLY WILL DISPLAY 'A 1 - '. PLACING THE SWITCH TO CHANNEL TWO ONLY WILL DISPLAY ' A 1 K'.

2. REQUIREMENTS:

HARDWARE: AA11-K INTERFACE WITH A SCOPE DISPLAY INSTALLED

STORAGE:: AAB REQUIRES:
1. DECIMAL WORDS: 453
2. OCTAL WORDS: 0705
3. OCTAL BYTES: 1612

3. PASS DEFINITION:

ONE PASS OF THE AAB MODULE CONSISTS OF DISPLAYING 55,296 POINTS ON THE SCREEN. THIS MEANS THAT 55,296 DATA TRANSFERS OCCURED ON THE UNIBUS.

4. EXECUTION TIME:

VARIES WITH SCOPE DELAY BUT SHOULD TAKE AN AVERAGE OF SIXTY SECONDS TO COMPLETE ONE PASS. WHEN RUNNING ALONE ON AN 11/05.

5. CONFIGURATION PARAMETERS:

DEFAULT PARAMETERS:

DVA: 170416, VCT: 360, BR1: 4

REQUIRED PARAMETERS:

NONE

6. DEVICE OPTION SETUP:

A. TURN ON SCOPE POWER.

B. PLACE CHANNEL SW TO 1 & 2 (IF VR14)

7. MODULE OPERATION:

7.1 TEST SEQUENCE:

A. START: USING THE DEVICE ADDRESS, THIS SECTION OF CODE, DETERMINES THE CONTROL, X AND Y POSITION ADDRESSES, AND VECTORS.

B. TSTNRG: THIS SECTION OF CODE PREFORMS A CONFIDENCE REGISTER TEST OF THE CONTROL, DAC 0, DAC 1, DAC 2, AND DAC 3 REGISTERS.

C. PRIME: IN THIS SECTION, THE FOUR DAC REGISTERS AND CONTROL REGISTERS ARE LOADED. THE SCOPE IS ENABLED AND AN 'EXIT' RETURN TO THE MONITOR.

D. AA11K: UPON A SCOPE INTERRUPT, THE PROGRAM WILL RETURN TO THIS CODE. ENTER DEFERRED SERVICE MODE AND TEST FOR A MODE FLAG. IF NO MODE FLAG, REPORT IT AS AN ERROR.

E. AA11KA: THRU CHAR13: THIS SECTION SELECTS
THE PROPER POINTS TO BE INTENSIFIED
ON THE SCREEN.

F. CHAR11: IN THIS CODE, THE CHANNEL BIT ALTERNATED
TO DISPLAY EACH CHANNEL.
IF A 611/613 SCOPE IS CONNECTED
ALTERNATING CHARACTERS WILL HAVE
ALTERNATING INTENSITY LEVELS.

G. CHAR20: IN THIS SECTION, THE PASS COUNT
IS DECREMENTED AND TESTED.
IF IT DID NOT BECOME ZERO, THEN
SELECT ANOTHER CLOCK RATE
AND RESUME COUNTING. UPON
A ZERO PASS COUNT THE CONTROL
AND PRESET REGISTER, ARE
CLEARED AND 'ENDPAS' IS
REPORTED.

8. OPERATOR OPTIONS:

A. LOCATION (VCPASS) CAN BE MODIFIED TO VARY THE NO.
LOOPS THRU TEST BEFORE END OF PASS IS REPORTED.

9. NON-STANDARD PRINTOUTS:

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED
IN THE DEC/X11 DOCUMENT

```
                                ;AA11-K  DEC/X11 EXERCISER MODULE

                                        IOMOD   <AABB >,170416,360,4,,,3000,123
                                        MODULE  140000,AABB ,170416,360,4,,,3000,123
                                        .TITLE  AABB DEC/X11 SYSTEM EXERCISER MODULE
                                ;       IDXCOM  VERSION   6       23-MAY-78
                                        .LIST   BIN
                                ;*****************************************************************************
000000'                         BEGIN:
000000' 040501  041102  040     MODNAM: .ASCII  /AABB / ;MODULE NAME.
000005'     000                 XFLAG:  .BYTE   0,0     ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 170416                  ADDR:   170416+0        ;1ST DEVICE ADDR.
000010' 000360                  VECTOR: 360+0           ;1ST DEVICE VECTOR.
000012'     200                 BR1:    .BYTE   PRTY4+0 ;1ST BR LEVEL.
000013'     000                 BR2:    .BYTE   PRTY4+0 ;2ND BR LEVEL.
000014' 000001                  DVID1:  +1              ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN            ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN            ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN            ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN            ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 140000                  STAT:   140000          ;STATUS WORD.
000030' 000244'                 INIT:   START           ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP           ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0               ;PASS COUNTER.
000036' 003000                  ICOUNT: 3000            ;# OF ITERATIONS PER PASS=3000
000040' 000000                  ICOUNT: 0               ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0               ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0               ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0               ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0               ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0               ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0               ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                 ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0               ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0               ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN            ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN            ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN            ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN            ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN            ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN            ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN            ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN            ;ADDR OF CURRENT CSR.
000102'                         SBADR:                  ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN            ;CONTENTS OF CSR.
000104'                         WASADR: OPEN            ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN            ;STATUS REG CONTENTS.
000106'                         ERRTYP:                 ;TYPE OF ERROR
000106' 000000                  ASR:    OPEN            ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN            ;ACTUAL DATA.
000112' 000416'                 RSTRT:  RESTRT          ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN            ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN            ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN            ;# OF INTERRUPTS PER ITERATION
```



```
000122' 000123                  IDNUM:  123             ;MODULE IDENTIFICATION NUMBER=123
        000040                          .REPT   SPSIZ   ;MODULE STACK STARTS HERE.
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
000224'                         MODSP:
                                ;*****************************************************************************
```

```
                                ;AA11-K OPERATOR CHANGEABLE LOCATIONS

000724' 000000                  VCPAS:  0


                                ; COMMON AA11-K DEVICE ADDRESSES

000226' 170416                  VCSTAT: 170416          ;SCOPE STATUS REGISTER
000230' 170420                  VCXREG: 170420          ;SCOPE X AXIS REGISTER
000232' 170422                  VCYREG: 170422          ;SCOPE Y AXIS REGISTER
000234' 170424                  VCDAC2: 170424          ;DAC #2 ADDRESS
000236' 170426                  VCDAC3: 170426          ;DAC #3 ADDRESS


                                ;COMMON AA11-K DEVICE VECTOR

000240' 000360                  VCIV:   360             ;SCOPE INTERRUPT VECTOR
000242' 000362                  VCIVS:  362

                                ;NOW SET UP THE ADDRESS AND VECTOR DISPATCH LOC.

000244' 016767  177566  177752  START:  MOV     ICONT,VCPAS     ;SAVE ICONT
000252' 005367  177746                  DEC     VCPAS           ;MUST BE ONE LESS
000256' 012767  000365  177634          MOV     #245.,INTR      ;245. INTERRUPTS/ITERATION
000264' 012767  000365  177622          MOV     #245.,WDTO      ;245. WORDS TO MEM/ITERATION
000272' 012767  000365  177616          MOV     #245.,WDFR      ;245 WORDS FROM MEM/ITERATION
000300' 016767  177502  177720          MOV     ADDR,VCSTAT
000306' 016767  177474  177714          MOV     ADDR,VCXREG
000314' 062767  000002  177706          ADD     #2,VCXREG
000322' 016767  177460  177702          MOV     ADDR,VCYREG
000330' 062767  000004  177674          ADD     #4,VCYREG
000336' 016767  177444  177670          MOV     ADDR,VCDAC2
000344' 062767  000006  177662          ADD     #6,VCDAC2
000352' 016767  177430  177656          MOV     ADDR,VCDAC3
000360' 062767  000010  177650          ADD     #10,VCDAC3
000366' 016767  177416  177644          MOV     VECTOR,VCIV
000374' 016767  177410  177640          MOV     VECTOR,VCIVS
000402' 062767  000002  177632          ADD     #2,VCIVS
000410' 016767  177612  177462          MOV     VCSTAT,CSRA     ;LOAD DEVICE ADDRESS
```



```
                                ;AA11-K LOGIC TEST

000416' 005077  177604  RESTRT: CLR     @VCSTAT         ;CLEAR STATUS
000422' 005077  177602          CLR     @VCXREG         ;CLEAR X
000426' 005077  177600          CLR     @VCYREG         ;CLEAR Y
000432' 005077  177576          CLR     @VCDAC2         ;CLEAR DAC #2
000436' 005077  177574          CLR     @VCDAC3         ;CLEAR DAC #3
000442' 005067  001114          CLR     FATAL           ;CLEAR FATAL ERROR FLAG

                                ;NOW TEST DAC#0 REGISTER BITS

000446' 016767  177556  177424  TSTXRG: MOV     VCXREG,CSRA     ;LOAD BUS ADDRESS
000454' 012767  010000  177422          MOV     #BIT12,ASTAT    ;LOAD EXPECTED BIT
000462' 006267  177416  1S:             ASR     ASTAT           ;CHANGE THE DATA
000466' 001422                          BEQ     2S              ;BR IF DONE
000470' 016777  177410  177532          MOV     ASTAT,@VCXREG   ;LOAD DAC # 0
000476' 017767  177526  177376          MOV     @VCXRFG,ACSR    ;READ DAC #
000504' 026767  177474  177370          CMP     ASTAT,ACSR      ;COMPARE RESULT
000512' 001763                          BEQ     1S              ;BR IF CORRECT
000514' 012767  000025  177364          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************
                                        ERRORS,BEGIN,NULL       ;DAC # 0 IN ERROR
                                        ;*****************************************************
000522' 104405  000000' 000000
000530' 005267  001026                  INC     FATAL           ;SET FATAL ERROR FLAG
000534' 000240                  2S:     NOP

                                ;NOW TEST DAC #1 REGISTER BITS

000536' 016767  177470  177334  TSTYRG: MOV     VCYREG,CSRA     ;LOAD BUS ADDRESS
000544' 012767  010000  177332          MOV     #BIT12,ASTAT    ;LOAD EXPECTED BIT
000552' 006267  177326  1S:             ASR     ASTAT           ;CHANGE THE DATA
000556' 001422                          BEQ     2S              ;BR IF DONE
000560' 016777  177320  177444          MOV     ASTAT,@VCYREG   ;LOAD DAC # 1
000566' 017767  177440  177306          MOV     @VCYREG,ACSR    ;READ DAC #
000574' 026767  177404  177300          CMP     ASTAT,ACSR      ;COMPARE RESULT
000602' 001763                          BEQ     1S              ;BR IF CORRECT
000604' 012767  000025  177274          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************
000612' 104405  000000' 000000          ERRORS,BEGIN,NULL       ;DAC # 1 IN ERROR
                                        ;*****************************************************
000620' 005267  000736                  INC     FATAL           ;SET FATAL ERROR FLAG
000624' 000240                  2S:     NOP
```

```
                                        ;NOW TEST DAC #2 REGISTER BITS
                                        ;
000626' 016767  177402  177244  TST2RG: MOV     VCDAC2,CSRA     ;LOAD BUS ADDRESS
000634' 012767  010000  177242          MOV     #BIT12,ASTAT    ;LOAD EXPECTED BIT
000642' 006267  177236          1S:     ASR     ASTAT           ;CHANGE THE DATA
000646' 001422                          BEQ     2S              ;BR IF DONE
000650' 016777  177230  177356          MOV     ASTAT,@VCDAC2   ;LOAD DAC # 2
000656' 017767  177352  177216          MOV     @VCDAC2,ACSR    ;READ DAC #
000664' 026767  177714  177210          CMP     ASTAT,ACSR      ;COMPARE RESULT
000672' 001763                          BEQ     1S              ;BR IF CORRECT
000674' 012767  000025  177204          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000702' 104405  000000' 000000          ERRORS,BEGIN,NULL       ;DAC # 2 IN ERROR
                                        ;*****************************************************************
000710' 005267  000646                  INC     FATAL           ;SET FATAL ERROR FLAG
000714' 000240                  2S:     NOP
                                        ;NOW TEST DAC #3 REGISTER BITS
                                        ;
000716' 016767  177314  177154  TST3RG: MOV     VCDAC3,CSRA     ;LOAD BUS ADDRESS
000724' 012767  010000  177151          MOV     #BIT12,ASTAT    ;LOAD EXPECTED BIT
000732' 006267  177146          1S:     ASR     ASTAT           ;CHANGE THE DATA
000736' 001422                          BEQ     2S              ;BR IF DONE
000740' 016777  177140  177270          MOV     ASTAT,@VCDAC3   ;LOAD DAC # 3
000746' 017767  177264  177126          MOV     @VCDAC3,ACSR    ;READ DAC #
000754' 026767  177724  177120          CMP     ASTAT,ACSR      ;COMPARE RESULT
000762' 001763                          BEQ     1S              ;BR IF CORRECT
000764' 012767  000025  177114          MOV     #25,ERRTYP      ;BIT STUCK
                                        ;*****************************************************************
000772' 104405  000000' 000000          ERRORS,BEGIN,NULL       ;DAC # 3 IN ERROR
                                        ;*****************************************************************
001000' 005267  000556                  INC     FATAL           ;SET FATAL ERROR FLAG
001004' 000240                  2S:     NOP
                                        ;
001006' 005767  000550                  TST     FATAL           ;TEST IF ANY FATAL LOGIC ERRORS
001012' 001402                          BEQ     PRIME
001014' 104410  000000'                 ENDS,BEGIN              ;DROP MODULE DUE TO DAC REGISTER ERROR
                                        ;
001020' 005077  177202          PRIME:  CLR     @VCSTAT         ;ENSURE CLEAR STATUS
001024' 012777  001052' 177206          MOV     #AA11K,@VCIV    ;SET UP LPSVC VECTOR
001032' 116777  176754  177202          MOVB    BR1,@VCTVS      ;SET UP
001040' 012777  000101  177160          MOV     #101,@VCSTAT    ;START DISPLAY AND INTERRUPT ENABLE
                                        ;
001046' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                                        ;DISPLAY        "AA11-K" ON THE SCOPE
                                        ;
001052'                         AA11K:
                                        ;-----------------------------------------------------------------
001052' 000004  000000' 001060'         PTRQS,BEGIN,1S          ; QUEUE UP TO CONTINUE AT 1S AND RTI
                                        ;-----------------------------------------------------------------
001060' 105777  177142          1S:     TSTB    @VCSTAT
001064' 100415                          BMI     AA11KA
001066' 017767  177134  177006          MOV     @VCSTAT,ACSR    ;LOAD VALUE
001074' 005077  177126                  CLR     @VCSTAT         ;
001100' 012767  000011  177000          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*****************************************************************
001106' 104405  000000' 000000          ERRORS,BEGIN,NULL       ; NO DISPLAY READY FLAG
                                        ;*****************************************************************
001114' 104410  000000'                 ENDS,BEGIN              ;
                                        ;
001120' 012767  002000  000436  AA11KA: MOV     #2000,XPOS      ;LOAD X AXIS
001126' 012767  005000  000432          MOV     #5000,YPOS      ;LOAD Y AXIS
001134' 012767  000006  000426          MOV     #6,CNTR         ;SET UP FOR 6 CHARACTERS
001142' 012767  001514' 000422          MOV     #TEXT,PNTR      ;TEXT AA11-K
001150' 012777  001276' 177062          MOV     #CHAR3,@VCIV    ;LOAD INTERRUPT VECTOR
001156' 017767  000410  000416  TXT1:   MOV     @PNTR,R1R2      ;LOAD
                                        ;PLOT CHARACTER
                                        ;
001164' 016767  000376  000402  CHAR:   MOV     YPOS,YPT        ;INIT POINT
001172' 042777  000016  177026          BIC     #16,@VCSTAT
001200' 016777  000360  177022          MOV     XPOS,@VCXREG
001206' 016777  000354  177016          MOV     YPOS,@VCYREG
001214' 052777  000010  177004  CHAR3:  BIS     #10,@VCSTAT     ;LOAD STATUS
001222' 012767  177773  000346          MOV     #-5,R1R0        ;MATRIX COUNT <ROW>
001230' 012767  177771  000342  CHAR1:  MOV     #-7,R1R1        ;MATRIX COUNT <COLUMN>
001236' 117767  000340  000340          MOVB    @R1R2,R1R3      ;GET CHARACTER
001244' 005267  000332                  INC     R1R2
001250' 106167  000330          CHAR2:  ROLB    R1R3
001254' 100033                          BPL     CHAR13
001256' 016777  000302  176744          MOV     XPOS,@VCXREG    ;LOAD X
001264' 016777  000276  176740          MOV     YPOS,@VCYREG
001272' 104400  000000'                 EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
001276'                         CHAR3:
                                        ;-----------------------------------------------------------------
001276' 000004  000000' 001304'         PTRQS,BEGIN,1S          ; QUEUE UP TO CONTINUE AT 1S AND RTI
                                        ;-----------------------------------------------------------------
001304' 105777  176716          1S:     TSTB    @VCSTAT
001310' 100415                          BMI     CHAR13
001312' 017767  176710  176562          MOV     @VCSTAT,R1R0    ;LOAD VALUE
001320' 005077  176702                  CLR     @VCSTAT         ;
001324' 012767  000011  176554          MOV     #11,ERRTYP      ;ILLEGAL INTERRUPT
                                        ;*****************************************************************
001332' 104405  000000' 000000          ERRORS,BEGIN,NULL       ;NO DISPLAY READY
                                        ;*****************************************************************
001340' 104410  000000'                 ENDS,BEGIN              ;
```

```
  423  001344' 062767  000070  000214  CHAR13: ADD     #70,XPOS        ;NEXT POINT
  424  001352' 005267  000222                  INC     AAR1            ;ALL POINTS IN THE COLUMN
  425  001356' 001334                          BNE     CHAR2
  426  001360' 016767  000210  000200          MOV     YPT,YPOS        ;LOAD NEXT COLUMN
  427  001366' 062767  000070  000170          ADD     #70,XPOS        ;ADD SCALE
  428  001374' 005267  000176                  INC     AAR0            ;DONE ALL COLUMN
  429  001400' 001313                          BNE     CHAR1
  430  001402' 062767  000070  000154          ADD     #70,XPOS        ;YES, NEXT CHARACTER
  431  001410' 062767  000002  000154          ADD     #2,CNTR
  432  001416' 032777  001000  176602  CHAR11: BIT     #BIT9,@VCSTAT   ;TEST CHANNEL
  433  001424' 001004                          BNE     1$
  434  001426' 052777  001000  176572          BIS     #BIT9,@VCSTAT   ;SET CHANNEL
  435  001434' 000403                          BR      CHAR20
  436  001436' 042777  030000  176562  1$:     BIC     #BIT,@VCSTAT    ;CLEAR CHANNEL
  437
  438  001444' 005367  000120          CHAR20: DEC     CNTR
  439  001450' 001017                          BNE     2$
  440  001452' 042777  000002  176546          BIC     #2,@VCSTAT
  441  001460' 026767  176540  176352          CMP     VCPAS,ICOUNT    ;TIME FOR FINAL ITERATION ?
  442  001466' 001404                          BEQ     1$              ;YES BRANCH
  443  001470' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                       ;MONITOR SHALL TEST END OF PASS
  444
  445  001474' 000167  177420                  JMP     AA11KA
  446  001500' 005077  176522          1$:     CLR     @VCSTAT
  447  001504' 104413  000000'                 ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                                       ;MONITOR SHALL TEST END OF PASS
  448
  449
  450  001510' 000167  177434          2$:     JMP     TXT1
```



```
  451                                      ;TEXT FOR THE AA11-K SCOPE OPTION
  452                                      ;TEXT =         "AA11-K"
  453
  454  001514' 001530'                     TXT:    A
  455  001516' 001530'                             A
  456  001520' 001554'                             N1
  457  001522' 001554'                             N1
  458  001524' 001547'                             DASH
  459  001526' 001535'                             K
  460
  461  001530'     176     021     021     A:      .BYTE   176,21,21,21,176
  462  001533'     021     176
  463  001535'     177     010     024     K:      .BYTE   177,10,24,42,101
  464  001540'     042     101
  465  001542'     046     111     111     S:      .BYTE   46,111,111,111,62
  466  001545'     111     062
  467  001547'     000     010     010     DASH:   .BYTE   0,10,10,10,0
  468  001552'     010     000
  469  001554'     000     102     177     N1:     .BYTE   0,102,177,100,0
  470  001557'     100     000
  471
  472          001562'                             .EVEN
  473
  474  001562' 000000                  FATAL:  0
  475  001564' 000000                  XPOS:   0
  476  001566' 000000                  YPOS:   0
  477  001570' 000000                  CNTR:   0
  478  001572' 000000                  PNTR:   0
  479  001574' 000000                  YPT:    0
  480  001576' 000000                  AAR0:   0
  481  001600' 000000                  AAR1:   0
  482  001602' 000000                  AAR2:   0
  483  001604' 000000                  AAR3:   0
  484  001606' 000000                  TEMP1:  0
  485  001610' 000000                  TEMP2:  0
  486
  487          000001                          .END
```

```
A       001530R      454     455     461#
AAR0    001576R      401*    428*    480#
AAR1    001600R      402*    424*    481#
AAR2    001602R      392*    403     404*    482#
AAR3    001604R      403*    405*    483#
AA11K   001052R      365     373#
AA11KA  001120R      378     387#    445
ACSR    000102R      224#    239*    300     316*    317     333*    334     350*    351     379*    416*
ADDR    000006R      190#    269     270     272     274     276
ADDR22= 001000       242#
ASH     000106R      228#
ASTAT   000104R      226#    295*    296*    298     300     312*    313*    315     317     329*    330*    332     334
                     346*    347*    349     351
AWAS    000110R      229#
BEGIN   000000R      187#    304     321     338     355     362     369     375     383     385     409     412     420
                     422     443     447
BIT0  = 000001       242#    244#
BIT1  = 000002       242#    244#
BIT10 = 002000       242#    244#
BIT11 = 004000       242#    244#
BIT12 = 010000       242#    244#    295     312     329     346
BIT13 = 020000       242#    244#
BIT14 = 040000       242#    244#
BIT15 = 100000       242#    244#
BIT2  = 000004       242#    244#
BIT3  = 000010       242#    244#
BIT4  = 000020       242#    244#
BIT5  = 000040       242#    244#
BIT6  = 000100       242#    244#
BIT7  = 000200       242#    244#
BIT8  = 000400       242#    244#
BIT9  = 001000       242#    244#    432     434
BREAKS= 104407       242#
BR1     000012R      192#    366
BR2     000013R      193#
BTODS = 104421       242#
CDATAS= 104412       242#
CHAR    001164R      396#
CHAR1   001230R      402#    429
CHAR11  001416R      432#
CHAR13  001344R      406     415     423#
CHAR2   001250R      405#    425
CHAR20  001444R      435     438#
CHAR3   001276R      391     410#
CHAR8   001214R      400#
CNTR    001570R      389*    438*    477#
CONFIG  000056R      212#
CSRA    000100R      222#    291*    294*    311*    328*    345*
DASH    001547R      456     467#
DATCKS= 104411       242#
DATERS= 104404       242#
DVI01   000014R      194#
ENDITS= 104413       242#    443     447
ENDS  = 104410       242#    362     385     422
ERRTYP  000105R      227#    302*    319*    336*    353*    381*    418*
EXITS = 104400       242#    369     409
```



```
FATAL   001562R      290*    306*    323*    340*    357*    360     474#
GETPAS= 104415       242#
GRUPS= 104414        242#
HRDCNT  000044R      207#
HRDERS= 104405       242#    304     321     338     355     383     420
HRDPAS  000050R      209#
ICONT   000036R      204#    264
ICOUNT  000040R      205#    441
IDNUM   000122R      234#
INIT    000030R      201#
IVTB    000120R      233#    266*
K       001536R      459     463#
MAP22S= 104416       242#
MODNAM  000000R      198#
MODSP   000224R      202     240#
MSGNS = 104403       242#
MSGSS = 104402       242#
MSGS  = 104401       242#
NULL  = 000000       242#    304     321     338     355     383     420
NI      001554R      455     457     469#
OPEN  = 000000       189     195     196     197     198     215     216     217     218     219     220     221     222
                     224     226     228     229     231     232     233     242#
OTOAS = 104420       242#
PASCNT  000034R      203#
PIRQS = 000004       242#    375     412
PNTR    001572R      390*    392     431*    478#
POPSP = 005726       242#
POPSP2= 022626       242#
PRIME   001020R      361     364#
PRTV  = 000000       193     242#
PRTV0 = 000000       242#
PRTV1 = 000040       242#
PRTV2 = 000100       242#
PRTV3 = 000140       242#
PRTV4 = 000200       192     242#
PRTV5 = 000240       242#
PRTV6 = 000300       242#
PRTV7 = 000340       242#
PS    = 177776       242#
PSW   = 177776       242#
PUSH  = 005046       242#
PUSH2 = 024646       242#
RANDS = 104417       242#
RANNUM  000054R      211#
RESTRT  000416R      230     285#
RES1    000056R      213#
RES2    000060R      214#
RSTPT   000112R      230#
S       001542R      465#
SBADR   000102R      223#
SOFCNT  000042R      206#
SOFERS= 104406       242#
SOFPAS  000046R      208#
SPOINT  000032R      202#
SPSIZ = 000040         1#    235
SR1     000016R      195#
```

```
SR2     000020R          196#
SR3     000022R          197#
SR4     000024R          198#
START   000244R          201     264#
STAT    000026R          200#
SVR0    000062R          215#
SVR1    000064R          216#
SVR2    000066R          217#
SVR3    000070R          218#
SVR4    000072R          219#
SVR5    000074R          220#
SVR6    000076R          221#
SVSCNF  000052R          210#
TEMP1   001606R          484#
TEMP2   001610R          485#
TEXT    001514R          390     454#
TRPDFD= 000022           242#
TSTXRG  000446R          294#
TSTYRG  000536R          311#
TST2RG  000626R          328#
TST3RG  000716R          345#
TXT1    001150R          391#    450
VCDAC2  000234R          254#    274*    275*    288*    328     332*    333
VCDAC3  000236R          255#    276*    277*    289*    345     349*    350
VCIV    000240R          259#    278*    365*    391*
VCIVS   000242R          260#    279*    280*    366*
VCPAS   000224R          246#    264*    265*    441
VCSTAT  000226R          247#    269*    281     285*    364*    367*    377     379     380*    397*    400*    414     416
                         417#    432     434*    436*    440*    446*
VCXREG  000230R          252#    270*    271*    286*    294     298*    299     398*    407*
VCYREG  000232R          253#    272*    273*    287*    311     315*    316     399*    408*
VECTOR  000010R          191#    278     279
WASADR  000104R          225#
WDFR    000116R          232#    268*
WDTO    000114R          231#    267*
XFLAG   000005R          189#
XPOS    001564R          387*    398     407     427*    430*    475#
YPOS    001566R          388*    396     399     408     423*    426*    476#
YPT     001574R          396*    426     479#
.       = 001612R        472#

. ABS.  000000      000
        001612      001

ERRORS DETECTED:  0
DEFAULT GLOBALS GENERATED:  0

XAAB80,XAAB80/SOL/CRF:SYM=DDXCOM,XAAB80
RUN-TIME:  1  1  .3 SECONDS
RUN-TIME RATIO:  11/3=3.1
CORE USED:  7K  (13 PAGES)
```

```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-F004C-MC

PRODUCT NAME: CXDRwC0 DR11-w MODULE

PRODUCT DATE: AUG 1982

MAINTAINER: DEC/X11 SUPPORT GROUP

AUTHOR: D. BUTENHOF

1. ABSTRACT

DRW IS AN IUMOD THAT EXERCISES ONE DR11-W. THE DEVICE IS EXERCISED USING THE MAINTENANCE MODE TO TRANSER A 16 WORD BUFFER.
MAINTENANCE MODE IS INTERNAL WRAP-AROUND LOGIC WHICH ELIMINATES THE NEED FOR A USER DEVICE OR PHYSICAL CABLE.

2. REQUIREMENTS

HARDWARE: ONE DR11-W INTERFACE
STORAGE:: DRW REQUIRES:
1. DECIMAL WORDS: 287
2. OCTAL WORDS: 0437
3. OCTAL BYTES: 1076

3. PASS DEFINITION

ONE PASS OF DRW CONSISTS OF TRANSFERRING ONE 16 WORD BLOCK OF DATA 77000(8) TIMES

4. EXECUTION TIME

ONE PASS OF DRW RUNNING ALONE ON A PDP11/04 PROCESSOR TAKES APPROXIMATELY 85 SECONDS.

5. CONFIGURATION REQUIREMENTS

DEFAULT PARAMETERS:

DEVADR: 172410, VECTOR: 124, BR1: 5, DEVCNT: 1

REQUIRED PARAMETERS:

NONE

6. DEVICE/OPTION SET-UP

INSTALL DR11-W (NO PHYSICAL CABLE IS NECESSARY)

7. MODULE OPERATION

TEST SEQUENCE:

A. SET UP VECTOR AND DEVICE REGISTERS
B. GENERATE READ-WRITE BUFFER
C. GENERATE CHECK BUFFER
D. TRANSFER 16 WORDS IN MAINT. MODE

E. COMPARE DATA IN:OUT - REPORT ERRORS
F. REPEAT 77000 TIMES
G. SIGNAL END OF PASS, RESTART AT A.

IF DEVICE FAILS TO INTERRUPT,AN ERROR MESSAGE WILL OCCUR AND THE PROGRAM WILL REQUEST THAT THE MODULE BE DROPPED.

8. OPERATION OPTIONS

NONE

9. NON STANDARD PRINTOUTS

NONE: ALL PRINTOUTS HAVE THE STANDARD FORMATS DESCRIBED IN THE DEC/X11 DOCUMENT.

10.0 READ-WRITE BUFFER DESCRIPTION

SUBROUTINE LODBUF LOADS DRBUF WITH AN INCREMENTING PATTERN (0,1,2,3,....15). WHEN XFERS ARE COMPLETE, DRBUF WILL BE MODIFIED(0,0,2,2,4,4,....14,14) DUE TO THE ALTERNATING DATI-DATO SEQUENCE CHARACTERISTIC OF DR11-W MAINTENANCE MODE.

11.0 CHECK BUFFER DESCRIPTION

SUBROUTINE LODCHK LOADS CHKBUF WITH MODIFIED PATTERN DESCRIBED IN 10.

12.0 BURST OPERATION

CHARACTERISTIC OF DR11-W MAINTENANCE MODE OPERATION, THE XFERS ARE MADE IN ALTERNATING SINGLE CYCLE & BURST MODE FASHION. FOR THE FIRST FOUR XFERS THE DR11-W REQUESTS THE BUS ONLY ONCE AND DOES NOT RELEASE IT UNTIL THE END OF THE FOURTH XFER. THE NEXT FOUR XFERS SEES THE DR11-W REQUESTING & RELEASING THE BUS FOR EACH XFER. THIS ALTERNATING ACTION CONTINUES FOR SUBSEQUENT XFERS.

```
;*************************************************************************
;*                                                                       *
;*  EDIT:   BY:             DATE:          REASON:                       *
;*--                                                                     *
;*  001     M DENSMORE      10-NOV-80      FIX LOADING OF CSR'S          *
;*--                                                                     *
;*  002     R GAUDIN        05-DEC-80      ADD EXIT & PIRQ CALLS.        *
;*                                         CHANGE "IST" OF CSR TO "CLR". *
;*                                         FIX LOAD OF "WAS" ERR PNTOUT. *
;*--                                                                     *
;*************************************************************************
```



```
000000            10MOD   <DRWC >,172410,124,5,0,0,77000,165                ;MJD001
000000            MODULE  140000,DRWC ,172410,124,5,0,0,77000,165,,,,,
                  .TITLE  DRWC DEC/X11 SYSTEM EXERCISER MODULE
                  ;       DDXCOM  VERSION  6.4     28-JAN-82
                                  LIST    MIM
                          ;*************************************************************************
000000                    BEGIN:
000000    104     122     127     MODNAM: .ASCII  /DRWC / ;MODULE NAME.
000003    103     040
000005    000                     XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006 172410                     ADDR:   172410+0                ;1ST DEVICE ADDR.
000010 000124                     VECTOR: 124+0                   ;1ST DEVICE VECTOR.
000012    240                     BR1:    .BYTE   PRTY5+0         ;1ST BR LEVEL.
000013    000                     BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014 000001                     DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016 000000                     SR1:    OPEN                    ;SWITCH REGISTER 1
000020 000000                     SR2:    OPEN                    ;SWITCH REGISTER 2
000022 000000                     SR3:    OPEN                    ;SWITCH REGISTER 3
000024 000000                     SR4:    OPEN                    ;SWITCH REGISTER 4
                          ;*************************************************************************
000026 140000                     STAT:   140000                  ;STATUS WORD.
000030 000226'                    INIT:   START                   ;MODULE START ADDR.
000032 000224'                    SPOINT: MODSP                   ;MODULE STACK POINTER.
000034 000000                     PASCNT: 0                       ;PASS COUNTER.
000036 077000                     ICONT:  77000                   ;# OF ITERATIONS PER PASS=77000
000040 000000                     ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042 000000                     SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044 000000                     HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046 000000                     SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050 000000                     HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052 000000                     SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054 000000                     RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056                            CONFIG:                         ;RESERVED FOR MONITOR USE
000056 000000                     RES1:   0                       ;RESERVED FOR MONITOR USE
000060 000000                     RES2:   0                       ;RESERVED FOR MONITOR USE
000062 000000                     SVR0:   OPEN                    ;LOC TO SAVE R0.
000064 000000                     SVR1:   OPEN                    ;LOC TO SAVE R1.
000066 000000                     SVR2:   OPEN                    ;LOC TO SAVE R2.
000070 000000                     SVR3:   OPEN                    ;LOC TO SAVE R3.
000072 000000                     SVR4:   OPEN                    ;LOC TO SAVE R4.
000074 000000                     SVR5:   OPEN                    ;LOC TO SAVE R5.
000076 000000                     SVR6:   OPEN                    ;LOC TO SAVE R6.
000100 000000                     CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102                            SBADR:                          ;ADDR OF GOOD DATA, OR
000102 000000                     ACSR:   OPEN                    ;CONTENTS OF CSR.
000104                            WASADR:                         ;ADDR OF BAD DATA, OR
000104 000000                     ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106                            ERRTYP:                         ;TYPE OF ERROR
000106 000000                     ASR:    OPEN                    ;EXPECTED DATA.
000110 000000                     AWAS:   OPEN                    ;ACTUAL DATA.
000112 000300'                    RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114 000000                     WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116 000000                     WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120 000000                     INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122 000165                     IDNUM:  165                     ;MODULE IDENTIFICATION NUMBER=165
        000040                            .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```

SEQ 5

```
                                        .NLIST
                                        .WORD   0
                                        .LIST
                                        .ENDR
     000224                             MODSP:
                                        ;*****************************************************************************
 175
 176                                    ;+
 177                                    ; THE NEXT LOCATION  CONTAINS A NUMBER  THAT CAN BE EXAMINED IN CORE TO  BE
 178                                    ; SURE THAT THE LISTING AND ACTUAL CODE ARE THE SAME.  THIS  LOCATION IS TO
 179                                    ; BE INCREMENTED  EACH TIME THE SOURCE CODE IS UPDATED.  LOCATION  'MODREV'
 180                                    ; IS NOT USED BY THE PROGRAM.
 181                                    ;-
 182
 183 000224  000001                     MODREV: 1+0
 184
 185                                    ;*****************************************************************************
```


SEQ 6

```
 187                                    ; * * *  START OF MODULE CODE  * * *
 188
 189                                    ;THIS MODULE TESTS THE DR-11W DIRECT MEMORY ACCESS INTERFACE
 190
 191                                            ;INITIALIZATION FOR (DMA) DR-11W
 192
 193
 194 000226                             START:
 195 000226  016705  177554                     MOV     ADDR,R5         ;GET 1ST ADDRESS
 196 000232  010567  000560                     MOV     R5,#CR          ;SET UP WORD COUNT REG.POINTER
 197 000236  005725                             TST     (R5)+           ;MAKE CERTAIN REG. CAN BE ACCESSED
 198 000240  010567  000554                     MOV     R5,BAR          ;SET UP BUS ADDRESS REG. POINTER
 199 000244  005725                             TST     (R5)+           ;ACCESS REG.
 200 000246  010567  000550                     MOV     R5,CSR          ;SET UP CONTROL STATUS REG. POINTER
 201 000252  005025                             CLR     (R5)+           ;ACCESS REGISTER                                ;RG02
 202 000254  010567  000544                     MOV     R5,DBR          ;SET UP DATA BUFFER REG. POINTER
 203 000260  005715                             TST     (R5)            ;ACCESS REG.
 204 000262  016705  177522                     MOV     VECTOR,R5       ;LOAD DEVICE VECTOR
 205 000266  010567  000534                     MOV     R5,DRVEC0       ;SETUP DEV VECTOR POINTER
 206 000272  005725                             TST     (R5)+           ;ACCESS REGISTER
 207 000274  010567  000530                     MOV     R5,DRVEC2       ;SETUP PRIORITY POINTER
 208
 209                                            ; DEVICE SERVICE CODE
 210
 211 000300  012767  077000  000472  RESTRT: MOV     #77000,INTPSC   ;TIMER FOR WHEN TO END PASS
 212 000306                             CONT:
     000306  104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR...
     000312  104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 213 000316  012777  177760  000472           MOV     #-20,@WCR       ;SIXTEEN WORD TRANSFER
 214 000324  012767  001032' 000456           MOV     #DRBUF,VA       ;SETUP BASE ADDRESS
 215 000332  104415  000000' 001010'           GETPA$,BEGIN, VA                ;GET PHYSICAL ADDRESS FROM 16-BIT VA
 216 000340  016777  000446  000452           MOV     PA,@BAR         ;SET UP REAL ADDR.
 217 000346  005067  000362                     CLR     SCSR            ;CLEAR THE SOFTWARE CSR                         ;MD01
 218 000352  016777  000356  000442           MOV     SCSR,@CSR       ;LOAD REAL CSR                                  ;MD01
 219 000360  012777  177777  000436           MOV     #-1,@DBR        ;SET UP DATA
 220 000366  056767  000422  000340           BIS     EA,SCSR         ;SET EXTENDED MEM. BITS                         ;MD01
 221 000374  016777  000334  000420           MOV     SCSR,@CSR       ;PUT IN REAL CSR                                ;MD01
 222 000402  012777  000542' 000416           MOV     #DRIH,@DRVEC0   ;SET VECTOR TO SERVICE ROUTINE
 223 000410  116777  177376  000412           MOVB    BR1,@DRVEC2     ;SET PRIORITY
 224 000416  005067  000360                     CLR     INTFLG          ;CLEAR INTERRUPT FLAG
 225 000422  004767  000260                     JSR     PC,LODBUF       ;GENERATE READ AND WRITE BUFFER
 226 000426  004767  000216                     JSR     PC,LODCHK       ;GENERATE CHECKING BUFFER
 227 000432  012767  000010  000346           MOV     #10,TMRCN1      ;SET UP INTERRUPT LOOP
 228 000440  005067  000340           TIME:   CLR     TMR             ;SETUP TIMER COUNTER
 229 000444  052767  010000  000262           BIS     #10000,SCSR     ;MAINTENANCE MODE                               ;MD01
 230 000452  052767  000501  000254           BIS     #501,SCSR       ;IE,CYCLE,AND GO                                ;MD01
 231 000460  016777  000250  000334           MOV     SCSR,@CSR       ;LOAD REAL CSR                                  ;MD01
 232
 233                                            ; WAIT FOR INTERRUPT CODE
 234                                            ; ADD EXIT MONITOR CALL                                                ;RG02
 235
 236 000466  104400  000000'                   EXIT$,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
 237
 238 000472                             TIMER:
     000472  104407  000000'                   BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR...
     000476  104407  000000'                   BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
 239 000502  005767  000274                     TST     INTFLG          ;DID INTERRUPT OCCUR?
```

```
                                        BEQ     1$                      ;IF NOT,CONTINUE IN TIMER
                                        JMP     CHECK                   ;IF YES,GO CHEC DATA
                                1$:     DEC     TMR                     ;COUNT MORE TIME;IS TMR FINISHED
                                        BNE     TIMER                   ;IF NOT,BREAK AGAIN
                                        DEC     TMRCNT                  ;DECREASE OVERALL COUNT
                                        BPL     TIMER                   ;GO AGAIN IF MORE TIME
                                        MSGN$,BEGIN,HUNG                ;ASCII MESSAGE CALL WITH COMMON HEADER
                                        END$,BEGIN                      ;

                                        ;INTERRUPT SERVICE CODE

                                DRIR:   INC     INTFLG                  ;INTERRUPT HAS OCCURRED
                                        CLR     @CSR                    ;CRL CSR REGISTER                                       ;RG02

                                        ; CHANGE RTI TO FIRQ MONITOR CALL                                                       ;RG02
                                        ;-------------------------------------------------------------------------------------
                                        FIRQ$,BEGIN,TIMER               ; QUEUE UP TO CONTINUE AT TIMER AND RTI
                                        ;-------------------------------------------------------------------------------------

                                        ;DATA CHECK CODE

                                CHECK:  MOV     #DRBUF,R1               ; GET THE BUFFER ADDRESS
                                        MOV     #8.,R2                  ; LOAD THE COUNTER
                                        MOV     #CHKBUF,R3              ;GET CHECK BUFFER ADDRESS
                                1$:     CMP     (R1)+,(R3)+             ; ARE THE TWO SEQUENTIAL WORDS EQUAL ?
                                        BNE     2$                      ; NO, REPORT AN ERROR
                                        DEC     R2                      ; DONE THE WHOLE BUFFER ?
                                        BGT     1$                      ; NO, KEEP CHECKING
                                        BR      DONE                    ;YES;CHECK FOR ENDPASS
                                2$:     MOV     CSR,ACSR                ;ADDRESS OF CONTROL STATUS REGISTER
                                        MOV     -(R3),AWAS              ;ACTUAL DATA                                            ;RG02
                                        MOV     R3,AWASADR              ;ADDRESS OF ACTUAL DATA                                 ;RG02
                                        MOV     -(R1),ASB               ;CORRECT DATA
                                        MOV     R1,SHADR                ;ADDRESS OF CORRECT DATA
                                        ;*********************************************************************
                                        DATER$,BEGIN                    ;DATA ERROR!!!
                                        ;*********************************************************************
                                DONE:
                                        END1$,BEGIN                     ;SIGNAL END OF ITERATION.
                                                                        ;MONITOR SHALL TEST END OF PASS
                                        JMP     CONT                    ;ELSE DO IT AGAIN

                                        ;-----------------------------
                                LODCHK: MOV     #CHKBUF,R2              ;LOAD CHKBUF WITH INCREMENTING
                                        CLR     R1                      ;MODIFIED PATTERN
                                        CLR     R3
                                1$:     MOV     R3,(R2)+
                                        MOV     R3,(R2)+
                                        ADD     #2,R1
                                        CMP     R1,#20                  ;ARE WE FINISHED?
                                        BNE     2$                      ;NO
                                        RTS     PC                      ;YES
                                2$:     ADD     #2,R3
                                        BR      1$

                                        ;-----------------------------
                                LODBUF: MOV     #DRBUF,R1               ;LOAD DRBUF WITH INCREMENTING PATTERN
```



```
                                        CLR     R2
                                        CLR     (R1)+
                                1$:     INC     R2
                                        CMP     R2,#20                  ;ARE WE DONE?
                                        BEQ     LDEXIT                  ;YES
                                        MOV     R2,(R1)+                ;NO,CONTINUE
                                        BR      1$
                                LDEXIT: RTS     PC

                                SCSR:   0                               ;SOFTWARE CSR                                           ;MD01
                                HUNG:   FAIL
                                        177777
                                FAIL:   .ASCIZ  /%DEVICE FAILED TO INTERRUPT%/
                                        .EVEN
                                INTPSC: OPEN                            ;COUNTS FOR WHEN TO ENDPASS
                                INTFLG: OPEN
                                TMR:    OPEN
                                TMRCNT: OPEN
                                VA:     OPEN
                                PA:     OPEN
                                EA:     OPEN
                                WCR:    OPEN
                                BAR:    OPEN
                                CSR:    OPEN
                                DBR:    OPEN
                                DRVEC0: OPEN
                                DRVEC2: OPEN
                                DRBUF:  .BLKW 20
                                CHKBUF: .BLKW   20
                                .END
```

```
DRWC DEC/X11 SYSTEM EXERCISER M MACRO M1200  02-SEP-82 14:16  PAGE 4-3
SYMBOL TABLE

ACSR     000102R          CHECK    000560R          ICOUNT   000040R          PDP70  = 010000          SOFERS= 104406
ADDR     000006R          CHKBUF   001072R          IDNUM    000122R          PIRQS  = 000004          SOFPAS  000046R
ADDR22= 001000            CKHNGS= 000001            INDPAR=  000040           POPSP  = 005726           SPOINT  000032R
APTPRE= 000200            CLKPRE= 000001            INIT     000030R          POPSP2= 022626            SPSIZ  = 000040
ASB      000106R          CLKSPS= 104422            INTFLG   001002R          PRHRS$= 000002            SR1      000016R
ASTAT    000104R          CONFIG   000056R          INTPSC   001000R          PRTY   = 000000           SR2      000020R
AUTO   = 000010           CONT     000306R          INTR     000120R          PRTY0  = 000000           SR3      000022R
AWAS     000110R          CSR      001022R          KIPRES= 000400            PRTY1  = 000040           SR4      000024R
BAR      001020R          CSRA     001000R          KTXTNO= 040000            PRTY2  = 000100           STAKT    000026R
BEGIN    000000R          DATCKS= 104411            LDEXIT   000732R          PRTY3  = 000140           STAT     000028R
BIT0   = 000001           DATERS= 104404            LODBUF   000706R          PRTY4  = 000200           SVR0     000062R
BIT1   = 000002           DBR      001024R          LODCHK   000650R          PRTY5  = 000240           SVR1     000064R
BIT10  = 002000           DONE     000640R          MAP22S= 104416            PRTY6  = 000300           SVR2     000066R
BIT11  = 004000           DRBUF    001032R          MODNAM   000000R          PRTY7  = 000340           SVR3     000070R
BIT12  = 010000           DRIR     000542R          MODREV   000224R          PS     = 177776           SVR4     000072R
BIT13  = 020000           DRVEC0   001026R          MODSP    000224R          PSW    = 177776           SVR5     000074R
BIT14  = 040000           DRVEC2   001030R          MSGN$  = 104403           PUSH   = 005746           SVR6     000076R
BIT15  = 100000           DVID1    000014R          MSGS$  = 104402           PUSH2  = 024646           SYSCNT   000052R
BIT2   = 000004           EA       001014R          MSG$   = 104401           PWRFLG= 000002            TIME     000440R
BIT3   = 000010           ECCMEM= 000100            NCPUDR= 000020            QMON22= 000010            TIMER    000472R
BIT4   = 000020           ENDITS= 104413            NOAPTY= 000002            RAND$  = 104417           TMR      001004R
BIT5   = 000040           ENDS   = 104410           NULL   = 000000           RANNUM   000054R          TMRCNT   001006R
BIT6   = 000100           ERRTYP   001016R          OPEN   = 000000           RESTRT   000300R          TRPDFD= 000023
BIT7   = 000200           EXITS  = 104400           OTOA$  = 104420           RES1     000056R          USTACK= 000001
BIT8   = 000400           FAIL     000742R          PA       001012R          RES2     000060R          VA       001010R
BIT9   = 001000           GETPAS= 104415            PARPRE= 002000            RH70   = 001000           VECTOR   000010R
BREAKS= 104407            GWBUFS= 104414            PASCNT   000034R          RSTRT    000112R          WASADR   000104R
BR1      000012R          HRDCNT   000044R          PATCH    001132R          R6     =%000006           WCH      001016R
BR2      000013R          HRDERS= 104405            PDPFIL= 000002            R7     =%000007           WDFR     000116R
BTOD$  = 104421           HRDPAS   000050R          PDPLSI= 020000            SPADP    000102R          WDID     000114R
CAPRES= 000004            HUNG     000736R          PDP44  = 100000           SCSR     000734R          XFLAG    000005R
CDATAS= 104412            ICONT    000036R          PDP60  = 004000           SOFCNT   000042R

. ABS.   000000     000
         001202     001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  13158 WORDS  ( 52 PAGES)
DYNAMIC MEMORY:  19748 WORDS  ( 75 PAGES)
ELAPSED TIME:  00:00:24
XDRWC0,XDRWC0/CR/-SP=DDXCUM,XDRWC0
```



```
XDRWC0     CREATED BY  MACRO  ON 2-SEP-82 AT 14:16     PAGE 1
SYMBOL CROSS REFERENCE                                  CREF   V01

SYMBOL  VALUE            REFERENCES

ACSR      000102 R       #3-174      *4-267
ADDR      000006 R       #3-174       4-195
ADDR22  = 001000         #3-174
APTPRE  = 000200         #3-174
ASB       000106 R       #3-174      *4-270
ASTAT     000104 P       #3-174
AUTO    = 000010         #3-174
AWAS      000110 R       #3-174      *4-268
BAR       001020 R       *4-198       4-216      #4-314
BEGIN     000000 R       #3-174       4-212       4-212       4-215       4-236       4-238       4-238       4-246       4-247
                          4-255       4-277       4-274
BIT0    = 000001         #3-174
BIT1    = 000002         #3-174
BIT10   = 002000         #3-174
BIT11   = 004000         #3-174
BIT12   = 010000         #3-174
BIT13   = 020000         #3-174
BIT14   = 040000         #3-174
BIT15   = 100000         #3-174
BIT2    = 000004         #3-174
BIT3    = 000010         #3-174
BIT4    = 000020         #3-174
BIT5    = 000040         #3-174
BIT6    = 000100         #3-174
BIT7    = 000200         #3-174
BIT8    = 000400         #3-174
BIT9    = 001000         #3-174
BREAKS  = 104407         #3-174       4-212       4-212       4-238       4-238
BR1       000012 R       #3-174       4-223
BR2       000013 R       #3-174
BTOD$   = 104421         #3-174
CAPRES  = 000004         #3-174
CDATAS  = 104412         #3-174
CHECK     000560 R        4-241      #4-259
CHKBUF    001072 R        4-261       4-278      #4-320
CKHNGS  = 000001         #3-174
CLKPRE  = 000001         #3-174
CLKSPS  = 104422         #3-174
CONFIG    000056 R       #3-174
CONT      000306 R       #4-212       4-275
CSR       001022 R       #4-200       4-218       4-221       4-231       4-252       4-267      #4-315
CSRA      001000 R       #3-174
DATCKS  = 104411         #3-174
DATERS  = 104404         #3-174       4-272
DBR       001024 R       *4-202       4-219      #4-316
DONE      000640 R        4-266      #4-273
DRBUF     001032 R        4-214       4-259       4-291      #4-319
DRIR      000542 R        4-222      #4-251
DRVEC0    001026 R       *4-205       4-222      #4-317
DRVEC2    001030 R       *4-207       4-223      #4-318
DVID1     000014 R       #3-174
EA        001014 R        4-220      #4-312
```

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES							
ECCMEM	= 000100		#3-174							
ENDITS	= 104413		#3-174	4-274						
ENDS	= 104410		#3-174	4-247						
ERRTYP	000106	R	#3-174							
EXITS	= 104400		#3-174	4-236						
FAIL	000742	R	4-302	#4-304						
GETPAS	= 104415		#3-174	4-215						
GWBUFS	= 104414		#3-174							
HRDCNT	000044	R	#3-174							
HRDERS	= 104405		#3-174							
HRDPAS	000050	R	#3-174							
HUNG	000736	R	4-246	#4-302						
ICUNT	000036	R	#3-174							
ICOUNT	000040	R	#3-174							
IDNUM	000122	R	#3-174							
INDPAR	= 000040		#3-174							
INIT	000030	R	#3-174							
INTFLG	001002	R	*4-224	4-239	*4-251	#4-307				
INTPSC	001000	R	#4-211	#4-306						
INTP	000120	R	#3-174							
KTPPES	= 000400		#3-174							
KTXTND	= 040000		#3-174							
LDEXIT	000732	R	4-296	#4-299						
LDBUF	000706	R	4-225	#4-291						
LDDCHK	000650	R	4-226	#4-278						
MAP22S	= 104416		#3-174							
MODNAM	000000	R	#3-174							
MODREV	000224	R	#3-183							
MODSP	000224	R	3-174	#3-174						
MSGN$	= 104403		#3-174	4-246						
MSGS$	= 104402		#3-174							
MSGS	= 104401		#3-174							
NCPUOP	= 000020		#3-174							
NOAPTY	= 000002		#3-174							
NULL	= 000000		#3-174							
OPEN	= 000000		3-174	3-174	3-174	3-174	3-174	3-174	3-174	3-174
			3-174	3-174	3-174	3-174	3-174	3-174	3-174	3-174
			3-174	3-174	#3-174	4-306	4-307	4-308	4-309	4-310	4-311
			4-312	4-313	4-314	4-315	4-316	4-317	4-318	
OTOA$	= 104420		#3-174							
PA	001012	R	4-216	#4-311						
PARPRE	= 002000		#3-174							
PASCNT	000034	R	#3-174							
PATCH	001132	R	#4-321							
PDPF11	= 000002		#3-174							
PDPLSI	= 020000		#3-174							
PDP44	= 100000		#3-174							
PDP60	= 004000		#3-174							
PDP70	= 010000		#3-174							
PIRQS	= 000004		#3-174	4-255						
POPSP	= 005726		#3-174							
POPSP2	= 022626		#3-174							

SYMBOL CROSS REFERENCE CREF V01

SYMBOL	VALUE		REFERENCES							
PRHMS$	= 000002		#3-174							
PRTY	= 000000		#3-174							
PRTY0	= 000000		3-174	#3-174						
PRTY1	= 000040		#3-174							
PRTY2	= 000100		#3-174							
PRTY3	= 000140		#3-174							
PRTY4	= 000200		#3-174							
PRTY5	= 000240		3-174	#3-174						
PRTY6	= 000300		#3-174							
PRTY7	= 000340		#3-174							
PS	= 177776		#3-174							
PS#	= 177776		#3-174							
PUSH	= 005746		#3-174							
PUSH2	= 024646		#3-174							
P#HFLG	= 000002		#3-174							
QMON22	= 000010		#3-174							
RAND$	= 104417		#3-174							
RANNUM	000054	R	#3-174							
RESTRT	000300	R	3-174	#4-211						
RES1	000056	R	#3-174							
RES2	000060	R	#3-174							
RH70	= 001000		#3-174							
RSTRT	000112	R	#3-174							
R6	=%000006		#3-174							
R7	=%000007		#3-174							
SHADR	000102	R	#3-174	*4-271						
SOSR	000734	R	*4-217	4-218	*4-220	4-221	*4-229	*4-230	4-231	#4-301
SOFCNT	000042	R	#3-174							
SOFERS	= 104406		#3-174							
SOFPAS	000046	R	#3-174							
SPOINT	000032	R	#3-174							
SPSIZ	= 000040		#1-28	3-174						
SR1	000016	R	#3-174							
SR2	000020	R	#3-174							
SR3	000022	R	#3-174							
SR4	000024	R	#3-174							
START	000226	R	3-174	#4-194						
STA1	000026	R	#3-174							
SVR0	000062	R	#3-174							
SVR1	000064	R	#3-174							
SVR2	000066	R	#3-174							
SVR3	000070	R	#3-174							
SVR4	000072	R	#3-174							
SVR5	000074	R	#3-174							
SVR6	000076	R	#3-174							
SYSCNT	000052	R	#3-174							
TIME	000440	R	#4-228							
TIMER	000472	R	#4-238	4-243	4-245	4-255				
TMR	001004	R	*4-224	*4-242	#4-308					
TMRCNT	001006	R	*4-227	*4-244	#4-309					
TRPDFD	= 000023		#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174
			#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174

SYMBOL VALUE REFERENCES

		#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174
		#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174
		#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174
		#3-174	3-174	3-174	#3-174	3-174	3-174	#3-174	3-174	3-174
		#3-174	3-174	3-174	#3-174					
DSTACK	= 000001	#3-174								
VA	001010 R	#4-214	4-215	#4-310						
VECTOR	000010 R	#3-174	4-204							
#ASADR	000104 R	#3-174	*4-269							
#CR	001016 R	*4-196	4-213	#4-313						
#DFR	000116 R	#3-174								
#DTD	000114 R	#3-174								
XFLAG	000005 R	#3-174								

MACRO NAME REFERENCES

#KMOD	#1-124								
#REAK	#1-222	4-212	4-238						
#TOD	#1-246								
CKDATA	#1-282								
CLKSP	#1-149								
DATACK	#1-291								
DATERR	#1-175	4-272							
DFSEVN	#1-314	3-174	3-174	3-174	3-174	3-174	3-174	3-174	3-174	3-174
	3-174	3-174	3-174							
DSEVNT	#1-324	3-174	3-174	3-174	3-174	3-174	3-174	3-174		
END	#1-212	4-247								
ENDIT	#1-203	4-274								
ENDMOD	#1-208									
EQUATS	#1-330	3-174								
EXIT	#1-157	4-236								
GETPA	#1-273	4-215								
GWBUFF	#1-261									
HLFBRK	#1-227									
HRDEK	#1-165									
IOMOD	#1-120	3-174								
IOMODP	#1-144									
IOMODR	#1-140									
IOMODX	#1-136									
MAP22	#1-277									
MODULE	#1-29	3-174								
MSG	#1-191									
MSGN	#1-195	4-246								
MSGS	#1-199									
NBKMOD	#1-132									
OTOA	#1-232									
PIRQ	#1-216	4-255								
RAND	#1-161									
SBKMOD	#1-128									

```
.REM _
```

IDENTIFICATION

PRODUCT CODE: AC-F001B-MC

PRODUCT NAME: CXKWFB0 GROSS TMNG MOD

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

"KWF" IS A GROSS TIMING ANALYSIS MODULE THAT USES THE KW11-L LINE CLOCK OPTION TO PERFORM AN OVERALL TIMING ANALYSIS ON THE 2040 PDP11 CONSOLE PROCESSOR SYSTEM OR A DN2X SECONDARY F.E. IT WILL BE CONFIGURED IN THE EXERCISER TO BE THE FIRST MODULE INITIALIZED BY THE MONITOR AND SINCE IT IS A "NRKMOD" IT WILL RUN ONLY ONE PASS AND NEVER RUN AGAIN UNTIL THE EXERCISER IS RESTARTED. ITS PURPOSE IS TO ESTABLISH CONFIDENCE IN THE HARDWARE AFFECTING OVERALL SYSTEM TIMING. IF THE TIMING IS FOUND TO BE WITHIN A PRE-ESTABLISHED, EMPIRICALLY DETERMINED LIMIT, THE MODULE WILL ENABLE USE OF THE SYSTEM TIMER MODULE. IF NOT IT WILL REPORT THE ERROR AND DISABLE USE OF THE SYSTEM TIMER MODULE.

2.0 REQUIREMENTS

HARDWARE: ANY PDP11/40 OR 11/34A SYSTEM WITH CORE MEMORY AND A KW11-L LINE CLOCK.

STORAGE:: KWF REQUIRES:
1. DECIMAL WORDS: 355
2. OCTAL WORDS: 0543
3. OCTAL BYTES: 1306

3.0 PASS DEFINITION

ONE PASS OF KWF CONSISTS OF A SINGLE PASS THROUGH THE MODULE CODE TO PERFORM THE REQUIRED DIAGNOSIS AND ANALYSIS.

4.0 EXECUTION TIME

ONE PASS OF KWF RUNNING ON AN 11/40 CPU TAKES APPROXIMATELY ONE MINUTE.

NOTE: THE TIMING INFORMATION IN THE "ENDPAS" MESSAGE FOR THE "KWF" MODULE IS MEANINGLESS SINCE THE TIMER MODULE DOES NOT GET STARTED UNTIL "KWF" REPORTS ITS END OF PASS.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 177546 VCT: 100 BR1: 6 BR2: 0 DVC: 1 SR1: 000037

REQUIRED PARAMETERS:

FOR 50 CYCLE SYSTEMS THE TIMING PARAMETER IN SR1 SHOULD BE SET TO 000030. (APPROX)

FOR CPU'S OTHER THAN THE KD11-A, THE VALUE OF SR1 WILL HAVE TO BE EMPIRACALLY DETERMINED BY RUNNING THIS MODULE ALONE AND NOTING THE INFORMATION PROVIDED IN THE ERROR PRINTOUT.

FOR AN 1134A 60 CYCLE SYSTEM THE TIMING PARAMETER IN SR1 SHOULD BE SET TO 000031.

6.0 DEVICE OPTION SET-UP

NONE REQUIRED

7.0 MODULE OPERATION

A. VERIFY THAT THE KW11-L MONITOR BIT CAN BE SET BY THE POWER SUPPLY SIGNAL. IF A FAILURE IS DETECTED, REPORT THE ERROR AND DROP THE MODULE.

B. VERIFY THAT THE KW11-L CAN GENERATE AN INTERRUPT TO THE PROPER VECTOR WHEN ENABLED. IF A FAILURE IS DETECTED REPORT THE ERROR AND DROP THE MODULE.

C. PERFORM THE TIMING ANALISIS:

1. SYNC THE LINE CLOCK
2. COUNT THE NO. OF ITERATIONS THROUGH A MONITOR "BREAK" LOOP WHILE WAITING FOR AN INTERRUPT.
3. STORE THE COUNT IN A TABLE
4. REPEAT (1) THRU (3) SIXTEEN TIMES TO STORE 16 COUNTS IN THE TABLE
5. SUM ALL ENTRIES IN THE TABLE AND DIVIDE BY 16 TO GET THE AVERAGE.
6. CHECK THAT THE AVERAGE IDS WITHIN LIMITS
7. IF WITHIN TOLERANCE, ENABLE SYSTEM TIMER TO BE ABLE TO RUN AND REPORT END OF PASS.
8. IF NOT WITHIN TOLERANCE, REPORT THE ERROR AND DROP THE MODULE. (SYSTEM TIMER NOT ENABLED TO RUN)

8.0 OPERATOR OPTIONS

"SR1" AND THE CONTENTS OF LOCATION "TOL" MAY HAVE TO BE MODIFIED FOR UNUSAL CASES TO COMPENSATE FOR A PARTICULAR SYSTEM INSTALLATION. [SR1] IS THE AVERAGE TIMER COUNT AND IF FOUND TO BE 000000 THE MODULE ASSUMES A DEFAULT OF "000037" (KD11-A CPU). IF NOT 000000, THE MODULE TAKES THE CONTENTS OF "SR1" AS THE AVERAGE TIME VALUE. THE LOCATION TAGGED "TOL" IS THE ALLOWABLE TOLERANCE AND IS PROGRAM LOADED AS A "3".

9.0 NON-STANDARD PRINTOUTS

IF A TIMING ANALYSIS ERROR IS DETECTED IT IS REPORTED
VIA AN EXTENDED ERROR PRINTOUT OF FOUR OCTAL NUMBERS:

WWWWWW SSSSSS HHHHHH LLLLLL

WHERE: WWWWWW IS THE WAS AVERAGE
        SSSSSS IS WHAT THE AVERAGE SHOULD HAVE BEEN
        HHHHHH IS THE HIGH LIMIT
        LLLLLL IS THE LOW LIMIT

```
000000'  NBKMOD  <KWFB >,177546,100,6,,,1,162
000000'           MODULE  41000,KWFB ,177546,100,6,,,1,162
                  .TITLE  KWFB DEC/X11 SYSTEM EXERCISER MODULE
                ;         DDXCOM  VERSION  6      23-MAY-78
                                  .LIST   BIN
                                  ;*****************************************************************************
000000'                           BEGIN:
000000' 053513  041106     040    MODNAM: .ASCII  /KWFB / ;MODULE NAME.
000005'     000                   XFLAG:  .BYTE   0PEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 177546                    ADDR:   177546+0                ;1ST DEVICE ADDR.
000010' 000100                    VECTOR: 100+0                   ;1ST DEVICE VECTOR.
000012'     300                   BR1:    .BYTE   PRTY6+0         ;1ST BR LEVEL.
000013'     000                   BR2:    .BYTE   PRTY+0          ;2ND BR LEVEL.
000014' 000001                    DVID1:  +1                      ;DEVICE INDICATOR 1.
000016' 000000                    SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                    SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                    SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                    SR4:    OPEN                    ;SWITCH REGISTER 4
                                  ;*****************************************************************************
000026' 041000                    STAT:   41000                   ;STATUS WORD.
000030' 000224'                   INIT:   START                   ;MODULE START ADDR.
000032' 000224'                   SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                    PASCNT: 0                       ;PASS COUNTER.
000036' 000001                    ICONT:  1                       ;# OF ITERATIONS PER PASS=1
000040' 000000                    ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                    SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                    HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                    SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                    HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                    SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                    RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                           CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                    RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                    RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                    SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                    SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                    SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                    SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                    SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                    SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                    SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                    CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                           SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                    ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                           WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                    ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                           ERRTYP:                         ;TYPE OF ERROR
000106' 000000                    ASR:    OPEN                    ;EXPECTED DATA.
000110' 000000                    AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000322'                   RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                    WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                    WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                    INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000162                    IDNUM:  162                     ;MODULE IDENTIFICATION NUMBER=162
        000040                            .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                          .NLIST
```



```
                                          .WORD   0
                                          .LIST
                                          .ENDR
000224'                           MODSP:
                                  ;*****************************************************************************
  232
  233                             .GLOBL  CLOCK,CLOCKL,CLOCKP,LCLEAR,PCLEAR       ;MONITOR AND TIMER MODULE LOCATIONS
  234
  235  000224' 132767  000001  000000G START:  BITB    #BIT0,CLOCK     ;USING A KW11-L TIMER ?
  236  000232' 001006                          BNE     1$              ;BR IF YES
  237  000234' 132767  000002  000000G         BITB    #BIT1,CLOCK     ;USING THE KW11-P TIMER ?
  238  000242' 001010                          BNE     2$              ;BR IF YES
  239  000244' 104410  000000'                 ENDS,BEGIN              ;
  240  000250' 016700  000000G         1$:     MOV     CLOCKL,R0       ;GET POINTER TO TIMING INFORMATION
  241  000254' 012767  000000G 000734          MOV     #LCLEAR,TCLEAR  ;SET PROPER JSR ADDR
  242  000262' 000405                          BR      3$              ;CONTINUE
  243  000264' 016700  000000G         2$:     MOV     CLOCKP,R0       ;GET POINTER TO TIMING INFORMATION
  244  000270' 012767  000000G 000720          MOV     #PCLEAR,TCLEAR  ;SET PROPER JSR ADDR
  245  000276' 005720                  3$:     TST     (R0)+           ;POINT TO LOCATION CONTAINING "TIME"
  246  000300' 005010                          CLR     (R0)            ;INIT RUN TIME = 000000
  247  000302' 004777  000710                  JSR     PC,@TCLEAR      ; CLEAR MODULE PASSTIME TABLE IN CLOCK MODULE
  248  000306' 005767  177504                  TST     SR1             ;DOES SR1 SPECIFY A TIME ?
  249  000312' 001003                          BNE     RESTRT          ;BR IF IT DOES
  250  000314' 012767  000037  177474          MOV     #37,SR1         ;DEFAULT TO 37(9) - KD11-A CPU
  251
  252  000322' 016767  177460  177550 RESTRT: MOV     ADDR,CSRA       ;SAVE KW11-L ADDRESS
  253  000330' 016700  177454'                 MOV     VECTOR,R0       ;GET VECTOR ADDRESS
  254  000334' 012720  001124'                 MOV     #KWINT,(R0)+    ;STEER KW INTR'S TO KWINT
  255  000340' 116710  177446                  MOVB    BR1,(R0)
  256  000344' 005077  177436                  CLR     @ADDR
  257
  258                             ;VERIFY THAT KW MONITOR BIT (BIT07) CAN BE SET BY POWER SUPPLY SIGNAL
  259                             ;-----------------------------------------------------------------
  260
  261  000350' 005005                  KWDT1:  CLR     R5              ;INIT TIMER
  262  000352'                         1$:
  263  000352' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
  264  000356' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
  265  000362' 005305                          DEC     R5              ;COUNT TIMER
  266  000364' 001372                          BNE     1$              ;BR UNTIL TIMER = 0
  267  000366' 022777  000200  177412          CMP     #200,@ADDR      ;DID MONITOR BIT SET?
  268  000374' 001415                          BEQ     KWDT2           ;BR IF YES
  269  000376' 017767  177404  177476          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
  270  000404' 012767  000025  177474          MOV     #25,ERRTYP      ;
                                  ;*****************************************************************************
  271  000412' 104405  000000' 000000          ERDES$,BEGIN,NULL       ;KW11-L MONITOR BIT WON'T SET.
                                  ;*****************************************************************************
  272  000420' 105067  000000G                 CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
  273  000424' 104410  000000'                 ENDS,BEGIN              ;
  274
  275                             ;VERIFY THAT KW11-L CAN GENERATE AN INTERRUPT PROPERLY
  276                             ;-----------------------------------------------------
  277
  278  000430' 005067  000566          KWDT2:  CLR     KWFLG1          ;CLEAR SOFTWARE INTR. FLAG
  279  000434' 005005                          CLR     R5              ;INIT TIMER
  280  000436' 052777  000100  177342          BIS     #100,@ADDR      ;SET LKCSR INTR ENAB
```

```
000444' 042777  000200  177334          BIC     #200,@ADDR      ;CLEAR MONITOR BIT
000452' 005767  000544          1$:     TST     KWFLG1          ;DID KW11-L INTR OCCUR
000456' 100425                          BMI     2$              ;BR IF YES
000460' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000464' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000470' 005305                          DEC     R5              ;COUNT TIMER
000472' 001367                          BNE     1$              ;BR IF NO TIMEOUT
000474' 017767  177306  177400          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
000502' 005077  177300                  CLR     @ADDR           ;CLEAR INTR ENAB
000506' 012767  000023  177372          MOV     #23,ERRTYP
                                        ;*****************************************************************
000514' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;KW11-L FAILED TO INTR ON TIME
                                        ;*****************************************************************
000522' 105067  000000G                 CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
000526' 104410  000000'                 ENDS,BEGIN              ;
000532' 032777  000100  177246  2$:     BIT     #100,@ADDR      ;DID INTR SERVICE CLEAR BIT 06?
000540' 001417                          BEQ     KWDT3           ;BR IF YES
000542' 017767  177240  177332          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
000550' 005077  177232                  CLR     @ADDR           ;CLEAR LKCSR
000554' 012767  000025  177324          MOV     #25,ERRTYP
                                        ;*****************************************************************
000562' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;KW11 INTR SERVICE FAILED TO CLR BIT06
                                        ;*****************************************************************
000570' 105067  000000G                 CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
000574' 104410  000000'                 ENDS,BEGIN              ;

                                ;GROSS TIMING ANALYSIS ROUTINE - PERFORMS GROSS CPU/MEMORY TIMING ANALYSIS
                                ;-------------------------------------------------------------------
                                ;USING THE KW11-L
                                ;----------------

000600' 004767  000332          KWDT3:  JSR     PC,CLRTAB       ;GO CLEAR LOOP COUNTERS TABLE
000604' 012701  001246'                 MOV     #CTRTAB,R1      ;R1 POINTS TO LOOP COUNTERS TABLE
000610' 005067  000406          KWSYNC: CLR     KWFLG1          ;INIT SOFTWARE INTR. FLAG
000614' 005000                          CLR     R0              ;INIT TIMER
000616' 052777  000100  177162          BIS     #100,@ADDR      ;ENABLE INTRS
000624' 005767  000372          1$:     TST     KWFLG1          ;INTR OCCUR YET?
000630' 100425                          BMI     2$              ;BR IF YES
000632' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000636' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000642' 005300                          DEC     R0              ;COUNT TIMER
000644' 001367                          BNE     1$              ;BR IF NO TIMEOUT
000646' 017767  177134  177226          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
000654' 005077  177126                  CLR     @ADDR           ;CLEAR OUT LKCSR
000660' 012767  000023  177220          MOV     #23,ERRTYP
                                        ;*****************************************************************
000666' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;KW11-L TIMEOUT
                                        ;*****************************************************************
000674' 105067  000000G                 CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
000700' 104410  000000'                 ENDS,BEGIN              ;
000704' 005067  000312          2$:     CLR     KWFLG1          ;INIT SOFTWARE INTR FLAG
000710' 005000                          CLR     R0              ;INIT LOOP COUNTER
000712' 052777  000100  177066          BIS     #100,@ADDR      ;ENABLE KW11-L INTRS
000720' 005767  000276          3$:     TST     KWFLG1          ;INTR OCCUR YET?
000724' 100425                          BMI     4$              ;BR IF YES
000726' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
```



```
000732' 104407  000000'                 BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000736' 005200                          INC     R0              ;COUNT ONE PASS THRU BREAK LOOP
000740' 001367                          BNE     3$              ;BR IF NO TIMEOUT
000742' 017767  177040  177132          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
000750' 005077  177032                  CLR     @ADDR           ;CLEAR OUT LKCSR
000754' 012767  000023  177124          MOV     #23,ERRTYP
                                        ;*****************************************************************
000762' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;KW11-L TIMEOUT
                                        ;*****************************************************************
000770' 105067  000000G                 CLRB    CLOCK           ;DISABLE ALL TIMING CHECKS
000774' 104410  000000'                 ENDS,BEGIN              ;
001000' 010021                  4$:     MOV     R0,(R1)+        ;SAVE LOOP COUNT IN TABLE
001002' 022701  001306'                 CMP     #CTRTAB+40,R1   ;END OF TABLE ??
001006' 001300                          BNE     KWSYNC          ;GO DO IT AGAIN
001010' 004767  000140                  JSR     PC,TIMAVG       ;GO COMPUTE TABLE AVERAGE
001014' 010003                          MOV     R0,R3           ;R3 CONTAINS AVERAGE COUNT
001016' 016701  176774                  MOV     SR1,R1          ;GET WHAT AVERAGE COUNT SHOULD BE
001022' 010102                          MOV     R1,R2           ;PUT IT IN R2 TOO
001024' 066701  000170                  ADD     TOL,R1          ;R1 CONTAINS HIGH LIMIT COUNT
001030' 166702  000164                  SUB     TOL,R2          ;R2 CONTAINS LOW LIMIT COUNT
001034' 160301                          SUB     R3,R1           ;LESS THAN HIGH LIMIT?
001036' 103406                          BCS     5$              ;BR IF NOT
001040' 160203                          SUB     R2,R3           ;GREATER THAN LOW LIMIT?
001042' 103404                          BCS     5$              ;BR IF NOT
001044' 104413  000000'                 ENDIT$,BEGIN            ;SIGNAL END OF ITERATION.
                                        ;MONITOR SHALL TEST END OF PASS
001050' 000167  177246                  JMP     RESTRT
001054' 105067  000000G         5$:     CLRB    CLOCK           ;DISABLE USING THE TIMER MODULE
001060' 060001                          ADD     R0,R1           ;RESTORE HIGH LIMIT
001062' 012704  001236'                 MOV     #TIMES,R4       ;POINT TO SAVE AREA
001066' 010024                          MOV     R0,(R4)+        ;SAVE WAS COUNTER AVERAGE
001070' 016724  176722                  MOV     SR1,(R4)+       ;SAVE SHOULD BE COUNTER AVERAGE
001074' 010124                          MOV     R1,(R4)+        ;SAVE HIGH LIMIT
001076' 010224                          MOV     R2,(R4)+        ;SAVE LOW LIMIT
001100' 017767  176702  176774          MOV     @ADDR,ACSR      ;SAVE CONTENTS OF LKCSR
001106' 005067  176774                  CLR     ERRTYP
                                        ;*****************************************************************
001112' 104405  000000' 001224'         HRDER$,BEGIN,KWMSG      ;PRINT COUNTER INFORMATION
                                        ;*****************************************************************
001120' 104410  000000'                 ENDS,BEGIN              ;

                                ;GENERAL PUROSE UTILITY ROUTINES
                                ;*******************************

                                ;INTERRUPT SERVICE ROUTINE
                                ;-------------------------

001124' 005077  176656          KWINT:  CLR     @ADDR           ;CLEAR OUT LKCSR
001130' 005167  000066                  COM     KWFLG1          ;SET SOFTWARE INTR. FLAG
001134' 000002                          RTI                     ;RETURN TO MAINLINE

                                ;ROUTINE TO CLEAR LOOP COUNTER TABLE
                                ;-----------------------------------

001136' 012701  001246'         CLRTAB: MOV     #CTRTAB,R1      ;POINT TO FIRST TABLE ENTRY
001142' 005021                  1$:     CLR     (R1)+           ;CLEAR ONE WORD
```

```
001144' 022701  001306'          CMP     #CTRTAB+40,R1   ;DONE 16 WORDS?
001150' 001374                   BNE     1$              ;BR IF NOT
001152' 000207                   RTS     PC              ;RETURN TO TIMING ANALYSIS ROUTINE

                                 ;ROUTINE TO COMPUTE AVERAGE LOOP COUNT
                                 ;-------------------------------------

001154' 012701  001246'  TIMAVG: MOV     #CTRTAB,R1      ;R1 POINTS TO 1ST TABLE ENTRY
001160' 005002                   CLR     R2              ;INIT 32 BIT ACCUMULATOR
001162' 005003                   CLR     R3
001164' 062102           1$:     ADD     (R1)+,R2        ;ADD TABLE ENTRY TO LOW ACCUM.
001166' 005503                   ADC     R3              ;ADD IN ANY CARRIES TO HI ACCUM.
001170' 022701  001306'          CMP     #CTRTAB+40,R1   ;SUMMED ALL 16 ENTRIES?
001174' 001373                   BNE     1$              ;BR IF NOT
001176' 005000                   CLR     R0              ;INIT R0 TO ACCUMULATE AVERAGE
001200' 162702  000020   2$:     SUB     #16.,R2         ;SUBTRACT CONSTANT 16(10)
001204' 005603                   SBC     R3              ;BORROW IF REQUIRED
001206' 103402                   BCS     3$              ;BR IF OVER SUBTRACT - (WE'RE DONE)
001210' 005200                   INC     R0              ;COUNT NO. OF TIMES 16 WENT IN
001212' 000772                   BR      2$              ;GO SUBTRACT ANOTHER 16(10)
001214' 000207           3$:     RTS     PC              ;RETURN - AVERAGE IN R0

                                 ;CONSTANTS, VARIABLES, AND MISCELLANEOUS DATA AREAS
                                 ;-------------------------------------------------

001216' 000000           TCLEAR: 0                       ;HOLDS SUBROUTINE ADDR.
001220' 000003           TOL:    3                       ;TOLERANCE FOR LOOP COUNTER AVG.
001222' 000000           KWFLG1: 0                       ;SOFTWARE INTERRUPT FLAG
001224' 001236'          KWMSG:  TIMES                   ;POINTS TO WAS AVERAGE
001226' 001240'                  TIMES+2                 ;POINTS TO SHOULD BE AVERAGE
001230' 001242'                  TIMES+4                 ;POINTS TO HIGH LIMIT
001232' 001244'                  TIMES+6                 ;POINTS TO LOW LIMIT
001234' 177777                   177777                  ;TABLE TERMINATOR

001236' 000004           TIMES:  .BLKW   4.              ;RESERVE FOUR WORDS FOR ERROR INFO.

                                 ;16. WORD TABLE TO STORE SIXTEEN LOOP COUNTS
                                 ;-------------------------------------------

001246' 000020           CTRTAB: .BLKW   16.

000001                   .END
```



```
ACSR    000102R          214#  269*  290*  300*  325*  342*  373*
ADDR    000006R          160#  252   256*  267   269   282*  283*  290   291*  298   300   301*  318*
                         325   326*  335*  342   343*  373   386*
ADDR22= 001000           232#
ASB     000106R          218#
ASTAT   000104R          216#
AWAS    000110R          219#
BEGIN   000000R          177#  239   263   264   272   275   286   287   294   297   304   307   321
                         322   329   332   338   339   346   349   363   376   378
BIT0  = 000001           232#  235   237
BIT1  = 000002           232#
BIT10 = 002000           232#
BIT11 = 004000           232#
BIT12 = 010000           232#
BIT13 = 020000           232#
BIT14 = 040000           232#
BIT15 = 100000           232#
BIT2  = 000004           232#
BIT3  = 000010           232#
BIT4  = 000020           232#
BIT5  = 000040           232#
BIT6  = 000100           232#
BIT7  = 000200           232#
BIT8  = 000400           232#
BIT9  = 001000           232#
BREAK$= 104407           232#  263   264   286   287   321   322   338   339
BR1     000012R          182#  255
BR2     000013R          183#
BTOD$ = 104421           232#
CDATA$= 104412           232#
CLOCK = ****** G         233#  235   237   274*  296*  306*  331*  348*  366*
CLOCKL= ****** G         233#  240
CLOCKP= ****** G         233#  243
CLRTAB  001136R          314   393#
CONFIG  000056R          202#
CSRA    000100R          212#  252*
CTRTAB  001246R          315   351   393   395   402   407   434#
DATCK$= 104411           232#
DATER$= 104404           232#
DVID1   000014R          184#
ENDIT$= 104413           232#  363
END$  = 104410           232#  239   275   297   307   332   349   378
ERRTYP  000106R          217#  270*  292*  302*  327*  344*  374*
EXIT$ = 104400           232#
GETPA$= 104415           232#
GWBUF$= 104414           232#
HRDCNT  000044R          197#
HRDERS= 104405           232#  272   294   304   329   346   376
HRDPAS  000050R          199#
ICONT   000036R          194#
ICOUNT  000040R          195#
IDNUM   000122R          224#
INIT    000030R          191#
INTR    000120R          223#
KWDT1   000350R          261#
KWDT2   000430R          268   280#
```

```
KWDT3   000600R          299     314#
KWFLG1  001232R          280*    284     316*    319     333*    336     387*    422#
KWINT   001124R          254     386#
KWMSG   001224R          276     423#
KWSYNC  000610R          316#    352
LCLEAR= ****** G         233#    241
MAP22$= 104416           232#
MODNAM  000004R          178#
MODSP   000224R          192     230#
MSG#$ = 104403           232#
MSGS$ = 104402           232#
MSG$  = 104401           232#
NULL    000000           232#    272     294     304     329     346
OPEN  = 000000           179     195     196     197     198     205     206     207     208     209     210     211     212
                         214     216     218     219     221     222     223     232#
OTOAS = 104420           232#
PASCNT  000034R          193#
PCLEAR= ****** G         233#    244
PIRQS = 000004           232#
POPSP   005726           232#
POPSP2= 022626           232#
PRTY  = 000000           183     232#
PRTY0 = 000000           232#
PRTY1 = 000040           232#
PRTY2 = 000100           232#
PRTY3 = 000140           232#
PRTY4 = 000200           232#
PRTY5 = 000240           232#
PRTY6 = 000300           182     232#
PRTY7 = 000340           232#
PS    = 177776           232#
PSW   = 177776           232#
PUSH    005746           232#
PUSH2 = 024646           232#
RANDS = 104417           232#
RANNUM  000054R          201#
RESTRT  000322R          220     249     252#    365
RES1    000056R          203#
RES2    000060R          204#
RSTRT   000112R          220#
SBADR   000102R          213#
SOFCNT  000042R          196#
SOFERS= 104406           232#
SOFPAS  000046R          198#
SPOINT  000032R          192#
SPSIZ = 000040             1     225
SR1     000016R          185#    248     250*    355     370
SR2     000020R          186#
SR3     000022R          187#
SR4     000024R          188#
START   000224R          191     235#
STAT    000026R          190#
SVR0    000062R          205#
SVR1    000064R          206#
SVR2    000066R          207#
SVR3    000070R          208#
```

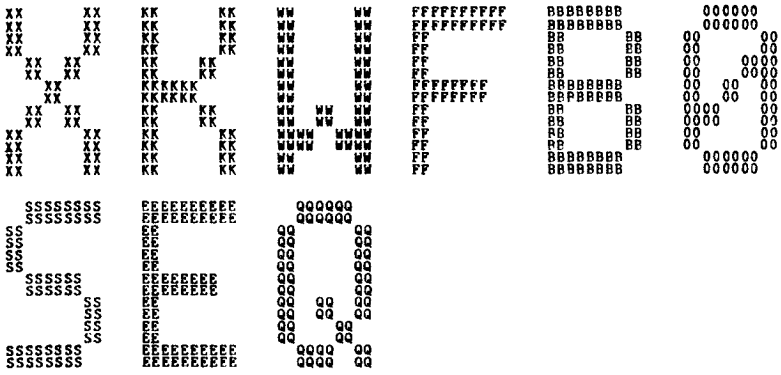


```
SVR4    000072R          209#
SVR5    000074R          210#
SVR6    000076R          211#
SYSCNT  000052R          200#
TCLEAR  001216R          241     244*    247     420#
TIMAVG  001154R          353     402#
TIMES   001236R          368     423     424     425     426     429#
TOL     001220R          357     358     421#
TRPDFD= 000022           232#
VECTOR  000010R          181#    253
WASADR  000104R          215#
WDFR    000116R          222#
WDTO    000114R          221#
XFLAG   000005R          179#
.     = 001306R          429#    434#

. ABS.  000000      000
        001306      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XKWFB0,XKWFB0/SOL/CRF:SYM=DDXCOM,XKWFB0
 RUN-TIME:  1  1  .2 SECONDS
 RUN-TIME RATIO: 26/3=7.7
 CORE USED:  7K  (13 PAGES)
```



```
**********************************************************************
* FOR REMOTE SITE PROBLEMS CONTACT THE I.P.C. NETWORK TEAM AT EX 5955.*
**********************************************************************
 *  *  *  *  *   L I S T I N G   P R O B L E M S ?   *  *  *  *  *  *
* If you have problems/questions with this listing, please            *
*  A. Check the applicable items.                                     *
*       ___ Do not need this listing, please discontinue              *
*       ___ Interested in receiving listing on microfiche             *
*       ___ Interested in receiving listing on 8 1/2 X 11 Xerox paper *
*       ___ Address is wrong or incomplete (Enter correctly below)    *
*       ___ Have other problem/question; not sure whom to contact     *
*  B. Identify yourself and the listing.          Phone:              *
*       Name:                          Location (mail stop):         *
*                                                                    *
*       CC #:       Listing name:                 Listing #:         *
*                                                                    *
*  C. Mail entire page to: Corporate Information Services  PK3-2/F34 *
 *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *  *

LPTSPL version 102(2263)/3(61) running on LPT061
 *START* User BEAUSOLEIL [400,2043] Job XKWFB0 Seq. 7423 Date 23-Oct-78 17:12:16 Monitor IPC-F 603 [6P8J *START*
/TO:ML21-4:BEAUSO -- Distribution to ML21-4, slot 132
File: COFF:XKWFB0.SEQ<057>[400,2043] Created: 12-Oct-78 16:48:00 Printed: 23-Oct-78 17:13:14
QUEUE Switches:  /PRINT:ARROW /FILE:ASCII /COPIES:2 /SPACING:1 /LIMIT:138 /FORMS:UNRDK
```

.9EM

IDENTIFICATION

PRODUCT CODE: AC-E998B-MC

PRODUCT NAME: CXDL890 DL11-E MODULE

PRODUCT DATE: SEPTEMBER 1978

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

DXDL8 IS AN IOMOD THAT EXERCISES ONE DL11-E ASYNCHRONOUS COMMUNICATIONS INTERFACE (W7800). THE PROGRAM CONSISTS OF TWO MAJOR SECTIONS AS DESCRIBED BELOW:

SECTION ONE:

THE FIRST SECTION CONSISTS OF A LOGICALLY SEQUENCED SET OF STATIC REGISTER TESTS TO VERIFY THE DL11-E HARDCORE REQUIRED TO PERFORM INPUT/OUTPUT DATA TRANSFERS IN INTERRUPT MODE. ERRORS DETECTED IN THIS SECTION THAT ARE DETERMINED TO BE FATAL ARE REPORTED VIA THE STANDARD DEC/X11 ERROR PRINTOUT AND THEN THE MODULE IS DROPPED FROM THE EXERCISE. NON-FATAL ERRORS ARE SIMPLY REPORTED AND THEN THE PROGRAM CONTINUES IN NORMAL SEQUENCE.

SECTION TWO:

THE SECOND SECTION TRANSFERS 256. BYTE BLOCKS OF DATA USING THE MAINTENANCE MODE TO TURN THE DATA AROUND. THE 256. BYTES OUTPUT ARE COMPARED WITH THE 256. BYTES INPUT FOR DATA COMPARISON ERRORS. ALL DATA COMPARISON ERRORS ARE REPORTED ON THE CONSOLE DEVICE. THE 256. BYTE TRANSFER IS REPEATED FOR FOUR DIFFERENT DATA BIT PATTERNS AS DESCRIBED BELOW:

A. NULL-DEL-NULL SEQUENCE (000,377,000.....,000,377)
B. BINARY UP-COUNT SEQUENCE (000,001,002.....,376,377)
C. BINARY DOWN COUNT SEQUENCE (377,376,375.....,001,000)
D. WORST CASE PATTERN (376,377,001,000.,000,200)

2.0 REQUIREMENTS

HARDWARE: A PDP11 COMPUTER WITH A DL11-E INTERFACE

STORAGE:: DL8 REQUIRES:
1. DECIMAL WORDS: 1354
2. OCTAL WORDS: 02512
3. OCTAL BYTES: 5224

3.0 PASS DEFINITION

ONE PASS OF "DXDL88" CONSISTS OF TWO ITERATIONS OF SECTION TWO OF THE MODULE CODE WHICH RESULTS IN 2048(10) BYTES TRANSFERRED.

4.0 EXECUTION TIME

AT 300 BAUD RUNNING ALONE ON A PDP11/40 A SINGLE ERROR FREE

PASS TAKES APPROXIMATELY 40. SECONDS THIS TIME WILL VARY DEPENDING UPON THE BAUD RATE AND CPU TYPE.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 175610 VCT: 300 BR1: 4 BR2: 0
DVC: 1 SR1: 0

REQUIRED PARAMETERS:

SR1 TO EXERCISE THOSE STATIC TESTS REQUIRING THE USE OF THE H315 MODEM TEST CONNECTOR (MODEM CONTROL LOGIC) BIT 15 OF SR1 MUST BE SET TO A "1". IE SR1=100000.

NOTE: IF SR1 BIT 15=1 AND THE MODEM TEST CONNECTOR IS NOT INSTALLED, FALSE ERRORS WILL BE REPORTED.

6.0 DEVICE/OPTION SETUP

IF THE MODEM CONTROL LOGIC IS TO BE TESTED, THE USER MUST DISCONNECT THE MODEM AND CONNECT THE H315 TEST CONNECTOR TO THE DL11-E DEVICE CABLE. SR1 MUST BE SETUP AS DESCRIBED IN (5.0) OR THE TESTS WILL BE SKIPPED.

7.0 MODULE OPERATION

7.1 TEST SEQUENCES

A. STATIC REGISTER TESTS

DLT01: TEST THAT ALL BITS IN THE RCSR ARE CLEAR WHEN THE MODULE IS INITIALIZED TO RUN.

DLT02: TEST THAT ONLY THE "READY" BIT IS SET IN THE XCSR WHEN THE MODULE IS INITIALIZED TO RUN.

DLT03: TEST THAT THE "MAINT" BIT IN THE XCSR CAN BE SET AND CLEARED.

DLT04: TEST THAT THE "INTR ENAB" BIT IN THE XCSR CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR WHEN SET AND ALSO THAT "INTR ENAB" CLEARS PROPERLY.

DLT05: TEST THAT A RECEIVER INTERRUPT OCCURS TO THE PROPER VECTOR WHEN "DONE" GETS SET WITH THE "INTR ENAB" BIT IN THE RCSR SET TO A ONE. ALSO TEST THAT THE CORRECT DATA IS RECEIVED.

TESTS DLT06 THRU DLT13 ASSUME THAT THE H315 MODEM

TEST CONNECTOR IS INSTALLED. THE USER INDICATES THIS BY SETTING BIT15 IN SR1. THE MODULE LOOKS AT SR1 AND WILL SKIP AROUND DLT06 THRU DLT13 IF BIT15=0.

DLT06: TEST THAT "REQ TO SEND" CAN ASSERT "RING" WHEN SET AND THAT BOTH "REQ TO SEND" AND "RING" CAN BE CLEARED PROPERLY.

DLT07: TEST THAT "SEC XMIT" WHEN SET ASSERTS "SEC REC" WHICH SETS "DATA SET INT" AND THAT READING THE RXSR CLEARS "DATA SET INT". ALSO TESTS THAT CLEARING "SEC XMIT" NEGATES "SEC REC" WHICH ALSO CAUSES "DATA SET INT" TO SET.

DLT10: TEST THAT "DTR" ASSERTS "CLR TO SEND" AND "CAR DET" WHICH IN TURN SET "DATA SET INT". ALSO TESTS THAT "CLR TO SEND" AND "CAR DET" CLEAR WHEN "DTR" IS CLEARED.

DLT11: TEST THAT "DATA SET INTR ENABLE" CAN BE SET AND CLEARED.

DLT12: TEST THAT "DATA SET INTR ENABLE" IN THE XCSR CAUSES AN INTR. WHEN ENABLED.

DLT13: TEST THAT THE BREAK BIT IN THE XCSR CAN BE SET AND CLEARED.

NOTE: BASIC TESTS DLT01 THRU DLT13 ARE EXECUTED ONLY ONCE WHEN THE MODULE IS FIRST INITIALIZED. IF ANY FATAL ERRORS ARE DETECTED THE MODULE IS DROPPED PRIOR TO THE DATA TRANSFER TESTS. AFTER PASS 1 THE MODULE IS RESTARTED AT THE ENTRY POINT TO THE DATA TRANSFER TESTS.

B. DATA TRANSFER TESTS

AFTER THE BASIC TESTS ARE RUN, FOUR 256(10) BYTE DATA TRANSFERS ARE EXECUTED IN THE MAINTENANCE MODE. EACH 256(10) BYTE BLOCK TRANSFER IS DIFFERENT IN THAT FOUR DIFFERENT DATA PATTERNS ARE XMITTED AND RECEIVED AS DESCRIBED IN PARA. 1.0.

THE TEST SEQUENCE FOR THE DATA TRANSFER TESTS IS AS FOLLOWS:

1.) CLEAR BOTH THE INPUT AND OUTPUT BUFFERS IN CORE (256(10) BYTES EACH).

2.) LOAD THE OUTPUT BUFFER WITH THE APPROPRIATE DATA PATTERN.

3.) ENABLE BOTH THE XMIT AND RCVR INTERRUPTS AND INITIATE THE DATA TRANSFERS.

SEQ 0005

4.) AFTER 256(10) BYTES HAVE BEEN OUTPUT AND INPUT COMPARE THE OUTPUT AND INPUT BUFFERS, BYTE BY BYTE FOR DATA COMPARE ERRORS. REPORT ALL DATA ERRORS ON THE CONSOLE DEVICE.

5.) IF ALL FOUR DATA PATTERNS HAVE BEEN TRANSFERRED, GO TO (6) BELOW - IF NOT REPEAT (1) THRU (4) FOR THE NEXT PATTERN.

6.) DECREMENT A PASS COUNTER (INITIALIAZED TO 2.) AND TEST FOR ZERO. IF ZERO GO TO (7) - IF NOT REPEAT (1) THRU (5) AGAIN.

7.) REPORT END OF PASS TO THE MONITOR AND RESTART AT (1) WITH THE FIRST DATA PATTERN.

NOTES:

(1) ON EACH "XMIT" INTERRUPT THE "READY" FLAG IS TESTED AND IF NOT SET, THE ERROR IS REPORTED AND THE MODULE IS DROPPED. (FALSE INTERRUPTS ARE CLASSIFIED AS FATAL ERRORS).

(2) ON EACH "RCVR" INTERRUPT THE "DONE" FLAG IS TESTED AND IF NOT SET THE MODULE IS DROPPED THE SAME AS FOR A "XMIT FALSE INTERRUPT".

(3) IF A SOFT ERROR (PARITY-FRAMING-OVERRUN) IS DETECTED IN RCVR INTR. SERVICE, THE OFFENDING BLOCK TRANSFER IS RESTARTED FROM THE BEGINNING OF THE BLOCK. IF AFTER THREE RETRIES THE ERROR PERSISTS, TRANSFER OF THE OFFENDING DATA PATTERN IS ABORTED AND THE PROGRAM GOES ON TO THE NEXT DATA PATTERN. ALL SOFT ERRORS ARE REPORTED ON THE CONSOLE DEVICE.

7.2 SUBROUTINE ABSTRACTS

SEGX: THIS SUBROUTINE SERVES AS A MINI-MONOTOR THAT CONTROLS THE SEQUENCING OF THE FOUR DIFFERENT 256(10) BYTE BLOCK TRANSFERS. IT IS CALLED AFTER THE BASIC TESTS AND PERFORMS THE FOLLOWING FUNCTIONS:

1. CALLS A SUBROUTINE TO CLEAR THE DATA BUFFERS
2. CALLS THE APPROPRIATE SUBROUTINE TO SET UP THE OUTPUT BUFFER WITH THE REQUIRED DATA PATTERN.
3. CALLS A SUBROUTINE TO ENABLE INTERRUPTS AND INITIATE THE DATA TRANSFER.
4. SERVICES RETRIES REQUESTED BY SOFT ERRORS.
5. PERFORMS "BREAK" CALLS TO THE MONITOR TO PREVENT TIMEOUTS FROM HANGING THE MODULE.
6. CALLS THE SUBROUTINE TO CHECK THE DATA BUFFERS WHEN THE BLOCK TRANSFER IS COMPLETE.

SEQ 0006

KICKOF: THIS SUBROUTINE IS CALLED FROM "SEGX" AND CONTAINS THE THE CODE TO ENABLE INTERRUPTS AND INITIATE THE BLOCK TRANSFER FOR EACH 256(10) BYTE BLOCK TRANSFER.

CHKDAT: THIS SUBROUTINE IS CALLED FROM "SEGX" AND CHECKS FOR DATA COMPARISON ERRORS AFTER EACH BLOCK TRANSFER.

STATR: THIS SUBROUTINE IS CALLED FROM THE BASIC TESTS AND SETS UP THE ERROR INFORMATION FOR ALL ERRORS RELATING TO THE RECEIVER CSR.

STATX: THIS ROUTINE IS CALLED FROM THE BASIC TESTS AND SETS UP THE ERROR INFORMATION FOR ALL ERRORS RELATING TO THE TRANSMITTER CSR.

CLDLBF: THIS ROUTINE IS CALLED FROM "SEGX" AND CLEARS BOTH THE OUTPUT AND INPUT DATA BUFFERS IN CORE.

LDOUT1: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH THE NULL-DEL-NULL PATTERN.

LDOUT2: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH A BINARY UP-COUNT PATTERN.

LDOUT3: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH A BINARY DOWN-COUNT PATTERN.

LDOUT4: THIS ROUTINE IS CALLED FROM "SEGX" AND IS USED TO LOAD THE OUTPUT BUFFER WITH THE MONOTOR'S WORST CASE PATTERN.

8.0 OPERATOR OPTIONS

A. USE THE MOD COMMAND TO MODIFY LOCATION "DLB 16" TO CHANGE SR1. REFER TO PARA. 5.0.

B. MODIFYING THE CONTENTS OF MODULE LOCATION "RESTRT +2" ALLOWS THE USER TO VARY THE TOTAL NO. OF BYTES TRANSFERED PER PASS. THIS IS DEFAULTED AT LOAD TIME TO 2 WHICH RESULTS IN 2048. BYTES TRANSFERRED.

9.0 NON-STANDARD ERROR PRINTOUTS

A. IF ANY ONE OF THE FOUR DATA PATTERNS OUTPUT CANNOT BE SUCCESSFULLY COMPLETED DUE TO SOFT ERRORS (3 RETRIES ATTEMPTED) OR A MONITOR "BREAK" TIMEOUT ONE OF THE FOLLOWING APPROPRIATE PRINTOUTS WILL OCCUR:

MSG1: "NULL-DEL-NULL SEQUENCE ABORTED"
MSG2: "BINARY UP-COUNT SEQUENCE ABORTED"
MSG3: "BINARY DOWN-COUNT SEQUENCE ABORTED

MSG4: "WORST CASE SEQUENCE ABORTED"

8. IF ANY FATAL ERROR DETECTED IN SECTION ONE RESULTS IN A DECISION TO DROP THE MODULE THE FOLLOWING MESSAGE IS PRINTED:

"FATAL ERROR DETECTED IN THE STATIC REGISTER TESTS"

AND THE "END" CALL IS EXECUTED TO DROP THE MODULE.


```
                ;DL11-E DEC/X11 EXERCISER MODULE

000000'                 IDWORD  <DLBR >,175610,300,4,,,1024.,<161
000000'                 MODULE  140000,DLBR ,175610,300,4,,,1024.,,161
                        .TITLE  DLBR DEC/X11 SYSTEM EXERCISER MODULE
                ;       DDXCOM  VERSION   6        23-MAY-78
                                .LIST     BIN
                        ;*****************************************************************
000000'                 BEGIN:
000000' 046104  041102      040 MODNAM: .ASCII  /DLBB / ;MODULE NAME.
000005'     000         XFLAG:  .BYTE   OPEN        ;USED TO KEEP TRACK OF MBUFF USAGE
000006' 175610          ADDR1:  175610+0            ;1ST DEVICE ADDR.
000010' 000300          VECTOR: 300+0               ;1ST DEVICE VECTOR.
000012'     200         BR1:    .BYTE   PRTY4+0     ;1ST BR LEVEL.
000013'     000         BR2:    .BYTE   PRTY+0      ;2ND BR LEVEL.
000014' 000001          DVID1:  +1                  ;DEVICE INDICATOR 1.
000016' 000000          SR1:    OPEN                ;SWITCH REGISTER 1
000020' 000000          SR2:    OPEN                ;SWITCH REGISTER 2
000022' 000000          SR3:    OPEN                ;SWITCH REGISTER 3
000024' 000000          SR4:    OPEN                ;SWITCH REGISTER 4
                        ;*****************************************************************
000026' 140000          STAT:   140000              ;STATUS WORD.
000030' 000224'         INIT:   START               ;MODULE START ADDR.
000032' 000224'         SPOINT: MODSP               ;MODULE STACK POINTER.
000034' 000000          PASCNT: 0                   ;PASS COUNTER.
000036' 002000          ICONT:  1024.               ;# OF ITERATIONS PER PASS=1024.
000040' 000000          ICOUNT: 0                   ;LOC TO COUNT ITERATIONS
000042' 000000          SOFCNT: 0                   ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000          HRDCNT: 0                   ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000          SOFPAS: 0                   ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000          HRDPAS: 0                   ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000          SYSCNT: 0                   ;# OF SYS ERRORS ACCUMULATED
000054' 000000          RANNUM: 0                   ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                 CONFIG:
000056' 000000          RRS1:   0                   ;RESERVED FOR MONITOR USE
000060' 000000          RRS2:   0                   ;RESERVED FOR MONITOR USE
000062' 000000          SVR0:   OPEN                ;LOC TO SAVE R0.
000064' 000000          SVR1:   OPEN                ;LOC TO SAVE R1.
000066' 000000          SVR2:   OPEN                ;LOC TO SAVE R2.
000070' 000000          SVR3:   OPEN                ;LOC TO SAVE R3.
000072' 000000          SVR4:   OPEN                ;LOC TO SAVE R4.
000074' 000000          SVR5:   OPEN                ;LOC TO SAVE R5.
000076' 000000          SVR6:   OPEN                ;LOC TO SAVE R6.
000100' 000000          CSRA:   OPEN                ;ADDR OF CURRENT CSR.
000102'                 SRADR:                      ;ADDR OF GOOD DATA, OR
000102' 000000          ACSR:   OPEN                ;CONTENTS OF CSR.
000104'                 #ASADR:                     ;ADDR OF BAD DATA, OR
000104' 000000          ASTAT:  OPEN                ;STATUS REG CONTENTS.
000106'                 ERRTYP:                     ;TYPE OF ERROR
000106' 000000          ASB:    OPEN                ;EXPECTED DATA.
000110' 000000          AWAS:   OPEN                ;ACTUAL DATA.
000112' 002334'         RSTRT:  RFSTRT              ;RESTART ADDRESS AFTER END OF PASS
000114' 000000          WDTO:   OPEN                ;WORDS TO MEMORY PER ITERATION
000116' 000000          WDFR:   OPEN                ;WORDS FROM MEMORY PER ITERATION
000120' 000000          INTR:   OPEN                ;# OF INTERRUPTS PER ITERATION
000122' 000161          IDNUM:  161                 ;MODULE IDENTIFICATION NUMBER=161
```

```
          000040                      .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
                                      .NLIST
                                      .WORD   0
                                      .LIST
                                      .ENDR
     000224'                  MODSP:
                              ;*******************************************************************
```



```
000224' 016700  177556          START:  MOV     ADDR,R0         ;GET BASE DEVICE ADDRESS
000230' 010067  003332                  MOV     R0,DLRCSR       ;SET UP RCVR CSR ADDRESS
000234' 005720                          TST     (R0)+
000236' 010067  003326                  MOV     R0,DLRDBR       ;SET UP RCVR DBR ADDRESS
000242' 005720                          TST     (R0)+
000244' 010067  003322                  MOV     R0,DLXCSR       ;SET UP XMITTR CSR ADDRESS
000250' 005720                          TST     (R0)+
000252' 010067  003316                  MOV     R0,DLXDBR       ;SET UP XMITTR DBR ADDRESS


                                ;               ******************************
                                ;               *   SECTION ONE              *
                                ;               ******************************

000256' 005077  003304          DLINIT: CLR     @DLRCSR         ;CLEAR OUT BOTH CSR'S
000262' 005077  003300                  CLR     @DLRCSR         ;MAKE SURE ALL DATA COMM BITS CLEARED
000266' 005077  003300                  CLR     @DLXCSR
000272' 005777  003272                  TST     @DLRDBR         ;FLUSH RCVR DONE BIT
000276' 005777  003266                  TST     @DLRDBR

                                ;THIS TEST VERIFIES THAT RCVR CSR GOT CLEARED UPON ENTRY
                                ;--------------------------------------------------------

000302' 005777  003260          DLT01:  TST     @DLRCSR         ;IS RCVR CSR ALL ZEROES ??
000306' 001415                          BEQ     DLT02           ;BR IF YES
000310' 004767  003052                  JSR     PC,STATR        ;GO SET UP ERROR INFO
000314' 012767  000025  177564          MOV     #25,ERRTYP
                                        ;*********************************************************
000322' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;CANT'T CLEAR OUT RCVR CSR
                                        ;*********************************************************
000330' 104403  000000' 003640'         MSGN$,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000336' 104410  000000'                 ENDS,BEGIN              ;

                                ;TEST THAT READY BIT IS ONLY BIT SET IN XMITTR CSR
                                ;--------------------------------------------------

000342' 022777  000200  003222  DLT02:  CMP     #200,@DLXCSR    ;READY SET ??
000350' 001415                          BEQ     DLT03           ;BR IF YES
000352' 004767  003026                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000356' 012767  000025  177522          MOV     #25,ERRTYP
                                        ;*********************************************************
000364' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;READY NOT SET OR OTHER BITS DIDN'T CLEAR IN XMIT CSR
                                        ;*********************************************************
000372' 104403  000000' 003640'         MSGN$,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000400' 104410  000000'                 ENDS,BEGIN              ;

                                ;TEST THAT MAINT. BIT CAN BE SET AND CLEARED IN XMIT CSR
                                ;--------------------------------------------------------

000404' 052777  000004  003160  DLT03:  BIS     #4,@DLXCSR      ;SET THE MAINT. BIT
000412' 022777  000204  003152          CMP     #204,@DLXCSR    ;DID IT SET ??
000420' 001415                          BEQ     1$              ;BR IF YES
000422' 004767  002756                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000426' 012767  000033  177452          MOV     #33,ERRTYP
                                        ;*********************************************************
```

```
000434' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MAINT BIT WON'T SET OR IT CLEARED READY
                                        ;*****************************************************************
000442' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000450' 104410  000000'                 ENDS,BEGIN              ;
000454' 042777  000004  003110  1S:     BIC     #4,@DLXCSR      ;NOW CLR THE MAINT BIT
000462' 022777  000200  003102          CMP     #200,@DLXCSR    ;DID IT CLEAR ??
000470' 001415                          BEQ     DLT04           ;BR IF YES
000472' 004767  002706                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000476' 012767  000025  177402          MOV     #25,ERRTYP
                                        ;*****************************************************************
000504' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;MAINT. BIT WON'T CLEAR OR READY CLEARED
                                        ;*****************************************************************
000512' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000520' 104410  000000'                 ENDS,BEGIN              ;


                                ;TEST THAT BIT 06 IN XCSR CAN CAUSE AN INTERRUPT
                                ;-----------------------------------------------
000524' 005067  003060          DLT04:  CLR     INTFLG          ;INIT THE SOFTWARE INTR. FLAG
000530' 016700  177254                  MOV     VECTOR,R0       ;GET BASE VECTOR ADDRESS
000534' 062700  000004                  ADD     #4,R0           ;GENERATE ADDR OF XMIT VECTOR
000540' 012720  000634'                 MOV     #2$,(R0)+       ;GO TO 2S ON XMIT INTERRUPT
000544' 116710  177242                  MOVB    BR1,(R0)        ;PRIORITY LEVEL = BR1
000550' 005001                          CLR     R1              ;INIT BREAK TIMER
000552' 052777  000100  003012          BIS     #100,@DLXCSR    ;SET INTR. ENAB
000560'                         1S:
000560' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
000564' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
000570' 005767  003014                  TST     INTFLG          ;DID XMIT INTR OCCUR YET ??
000574' 001025                          BNE     3S              ;BR IF IT DID
000576' 005301                          DEC     R1              ;COUNT BREAK TIMER
000600' 001367                          BNE     1S              ;BR IF NO TIMEOUT
000602' 004767  002576                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000606' 012767  000023  177272          MOV     #23,ERRTYP
                                        ;*****************************************************************
000614' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;XMITTER FAILED TO GENERATE INTERRUPT
                                        ;*****************************************************************
000622' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000630' 104410  000000'                 ENDS,BEGIN              ;
000634' 042777  000100  002730  2S:     BIC     #100,@DLXCSR    ;DISABLE XMITTR INTR ENABLE
000642' 005167  002742                  COM     INTFLG          ;SET THE INTR. FLAG
000646' 000002                          RTI                     ;RETURN CONTROL TO OTHER GUY
000650' 022777  000200  002714  3S:     CMP     #200,@DLXCSR    ;DID I.E. GET CLEARED IN INTR. SERVICE
000656' 001415                          BEQ     DLT05           ;BR IF YES
000660' 004767  002520                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000664' 012767  000027  177214          MOV     #27,ERRTYP
                                        ;*****************************************************************
000672' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"DONE" OR RCVR INTR. ENAB FAILED TO CLEAR
                                        ;*****************************************************************
000700' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000706' 104410  000000'                 ENDS,BEGIN              ;


                                ;TEST THAT A RCVR INTR CAN OCCUR WHEN "DONE" GETS SET
                                ;----------------------------------------------------
000712' 005067  002672          DLT05:  CLR     INTFLG          ;INIT SOFTWARE INTR. FLAG
```



```
000716' 016700  177066                  MOV     VECTOR,R0       ;GET THE BASE VECTOR ADDRESS
000722' 012720  001162'                 MOV     #4$,(R0)+       ;GO TO 4S ON RCVR INTERRUPT
000726' 116710  177060                  MOVB    BR1,(R0)        ;SET PRIORITY
000732' 052777  000100  002626          BIS     #100,@DLRCSR    ;SET I.E. IN RCVR CSR
000740' 032777  000100  002620          BIT     #100,@DLRCSR    ;DID IT SET
000746' 001015                          BNE     1S              ;BR IF IT DID
000750' 004767  002412                  JSR     PC,STATX        ;GO SET UP ERROR INFO
000754' 012767  000033  177124          MOV     #33,ERRTYP
                                        ;*****************************************************************
000762' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET BIT 06 IN RCSR I.E.
                                        ;*****************************************************************
000770' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
000776' 104410  000000'                 ENDS,BEGIN              ;
001002' 042777  000100  002556  1S:     BIC     #100,@DLRCSR    ;NOW CLEAR THE I.E. BIT
001010' 032777  000100  002550          BIT     #100,@DLRCSR    ;DID I.E. BIT GET CLEARED ??
001016' 001415                          BEQ     2S              ;BR IF YES
001020' 004767  002342                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001024' 012767  000023  177054          MOV     #23,ERRTYP
                                        ;*****************************************************************
001032' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T CLEAR RCSR INTR. ENAB. BIT
                                        ;*****************************************************************
001040' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001046' 104410  000000'                 ENDS,BEGIN              ;
001052' 052777  000100  002506  2S:     BIS     #100,@DLRCSR    ;NOW TURN IT ON FOR REAL
001060' 052777  000004  002504          BIS     #4,@DLXCSR      ;TURN ON MAINT. MODE
001066' 005001                          CLR     R1              ;INIT BREAK TIMER
001070' 112777  000252  002476          MOVB    #252,@DLXDBR    ;LOAD THE XMITTR OUTPUT DATA BUFFER
001076'                         3S:
001076' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
001102' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
001106' 005767  002476                  TST     INTFLG          ;DID RCVR INTR. YET ??
001112' 001036                          BNE     5S              ;BR IF IT DID
001114' 005301                          DEC     R1              ;COUNT BREAK TIMER
001116' 001367                          BNE     3S              ;BR IF NO TIMEOUT
001120' 004767  002242                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001124' 005077  002436                  CLR     @DLRCSR         ;CLEAR BOTH CSR'S
001130' 005077  002436                  CLR     @DLXCSR
001134' 012767  000023  176744          MOV     #23,ERRTYP
                                        ;*****************************************************************
001142' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RCVR FAILED TO INTR ON TIME
                                        ;*****************************************************************
001150' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001156' 104410  000000'                 ENDS,BEGIN              ;
001162' 117767  002402  176720  4S:     MOVB    @DLRDBR,AWAS    ;GET THE RECEIVED DATA
001170' 042777  000100  002370          BIC     #100,@DLRCSR    ;TURN OFF I.E.
001176' 005077  002370                  CLR     @DLXCSR         ;TURN OFF MAINTENANCE MODE
001202' 005167  002402                  COM     INTFLG          ;SET SOFTWARE INTR. FLAG
001206' 000002                          RTI                     ;RETURN TO OTHER GUY
001210' 005777  002352          5S:     TST     @DLRCSR         ;DID INTR SERVICE CLEAR THE RCVR CSR ??
001214' 001421                          BEQ     6S              ;BR IF IT DID
001216' 004767  002144                  JSR     PC,STATX        ;GO SET UP ERROR INFO
001222' 005077  002340                  CLR     @DLRCSR         ;CLEAR BOTH CSR'S
001226' 005077  002340                  CLR     @DLXCSR
001232' 012767  000025  176646          MOV     #25,ERRTYP
                                        ;*****************************************************************
001240' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RCVR INTR SERVICE FAILED TO CLEAR I.E. AND DONE
```

```
                                        ;*****************************************************************
001246' 104403  000000' 003640'         MSGNS,BEGIN,DRPMS       ;ASCII MESSAGE CALL WITH COMMON HEADER
001254' 104410  000000'                 ENDS,BEGIN              ;
001260' 122767  000252  176622  6S:     CMPB    #252,AWAS       ;WAS DATA CORRECT ???
001266' 001421                          BEQ     MODEM           ;BR IF YES
001270' 016767  002272  176602          MOV     DLRCSR,CSRA     ;SET UP ERROR INFO
001276' 017767  002264  176576          MOV     @DLRCSR,ACSR
001304' 012767  000252  176574          MOV     #252,ASR
001312' 016767  002252  176564          MOV     DLRDBR,WASADR
001320' 016767  002250  176554          MOV     DLXDBR,SBADR
                                        ;*****************************************************************
001326' 104404  000000'                 DATERS,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************


                                    ;THE FOLLOWING GROUP OF BASIC TESTS ASSUME THAT THE MODEM IS DISCONNECT-
                                    ;ED FROM THE DLII-E AND AN H315 MODEM TEST CONNECTOR IS CONNECTED IN ITS
                                    ;PLACE. ALL OF THESE TESTS ARE SKIPPED IF SR1(15) IS CLEARED.


001332' 005767  176460          MODEM:  TST     SR1             ;IS SR1 BIT 15 = 1 ??
001336' 100402                          BMI     DLT06           ;DO MODEM TESTS IF SET
001340' 000167  000770                  JMP     RESTRT          ;SKIP MODEM TESTS IF SR1(15)=0


                                    ;TEST THAT "REQ TO SEND" CAN ASSERT "RING"
                                    ;-----------------------------------------


001344' 052777  000004  002214  DLT06:  BIS     #4,@DLRCSR      ;SET REQ TO SEND
001352' 032777  000004  002206          BIT     #4,@DLRCSR      ;DID IT SET ??
001360' 001010                          BNE     1$              ;BR IF YES
001362' 004767  002000                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001366' 012767  000025  176512          MOV     #25,ERRTYP
                                        ;*****************************************************************
001374' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CAN'T SET REQ TO SEND
                                        ;*****************************************************************
001402' 032777  040000  002156  1S:     BIT     #40000,@DLRCSR  ;DID "RING" GET ASSERTED ??
001410' 001010                          BNE     2$              ;BR IF YES
001412' 004767  001750                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001416' 012767  000025  176462          MOV     #25,ERRTYP
                                        ;*****************************************************************
001424' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"REQ TO SEND" DIDN'T ASSERT "RING"
                                        ;*****************************************************************
001432' 042777  000004  002126  2S:     BIC     #4,@DLRCSR      ;TURN OFF "REQ TO SEND"
001440' 005777  002122                  TST     @DLRCSR         ;ARE ALL BITS NOW CLEAR ??
001444' 001410                          BEQ     DLT07           ;BR IF BOTH "RING" AND "REQ TO SEND " CLEARED
001446' 004767  001714                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001452' 012767  000025  176426          MOV     #25,ERRTYP
                                        ;*****************************************************************
001460' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"RING" OR "REQ TO SEND" FAILED TO CLEAR
                                        ;*****************************************************************


                                    ;TEST THAT "SEC XMIT" ASSERTS "SEC REC" WHICH SETS "DATA SET INT"
                                    ;---------------------------------------------------------------


001466' 052777  000010  002072  DLT07:  BIS     #10,@DLRCSR     ;SET SEC XMIT
001474' 005777  002066                  TST     @DLRCSR         ;DID DATA SET INT GET SET ??
001500' 100410                          BMI     1S              ;BR IF YES
001502' 004767  001660                  JSR     PC,STATR        ;GO SET UP ERROR INFO
```



```
001506' 012767  000025  176372            MOV     #25,ERRTYP
                                        ;*****************************************************************
001514' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DATA SET INT FAILED TO SET
                                        ;*****************************************************************
001522' 022777  002010  002036  1S:     CMP     #2010,@DLRCSR   ;SEC XMIT AND REC SET - DATA SET INT CLEAR
001530' 001410                          BEQ     2$              ;BR IF YES
001532' 004767  001630                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001536' 012767  000025  176342          MOV     #25,ERRTYP
                                        ;*****************************************************************
001544' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SEC XMIT OR SEC REC FAILED TO SET - DATA SET INT FAILED
                                        ;*****************************************************************
001552' 042777  000010  002006  2S:     BIC     #10,@DLRCSR     ;TURN OFF SEC XMIT
001560' 005777  002007                  TST     @DLRCSR         ;DID SEC XMIT GOING OFF SET DATA SET INT ?
001564' 100410                          BMI     3$              ;BR IF YES
001566' 004767  001574                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001572' 012767  000020  176306          MOV     #20,ERRTYP
                                        ;*****************************************************************
001600' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;CLEARING SEC XMIT FAILED TO SET DATA SET INT
                                        ;*****************************************************************
001606' 005777  001754          3S:     TST     @DLRCSR         ;ALL BITS NOW CLEAR
001612' 001410                          BEQ     DLT10           ;BR IF YES
001614' 004767  001546                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001620' 012767  000020  176260          MOV     #20,ERRTYP
                                        ;*****************************************************************
001626' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;SEC XMIT,SEC REC, OR DATA SET INT FAILED TO CLEAR
                                        ;*****************************************************************


                                    ;TEST THAT "DTR" ASSERTS "CAR DET", "CLR TO SEND", AND "DATA SET INT"
                                    ;-------------------------------------------------------------------


001634' 005077  001726          DLT10:  CLR     @DLRCSR         ;CLR THE RCVR CSR
001640' 052777  000002  001720          BIS     #2,@DLRCSR      ;SET DATA TERM READY
001646' 005777  001714                  TST     @DLRCSR         ;DID DATA SET INT SET ??
001652' 100407                          BMI     1S              ;BR IF YES
001654' 004767  001506                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001660' 005067  176222                  CLR     ERRTYP
                                        ;*****************************************************************
001664' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"DTR" FAILED TO SET "DATA SET INT"
                                        ;*****************************************************************
001672' 022777  030002  001666  1S:     CMP     #30002,@DLRCSR;DTR,CLR TO SEND, CAR DET, SET AND DATA SET INT CLEAR ?
001700' 001407                          BEQ     2$              ;BR IF YES
001702' 004767  001460                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001706' 005067  176174                  CLR     ERRTYP
                                        ;*****************************************************************
001712' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DTR,CAR DET, OR CLR TO SEND FAILED TO SET OR DATA SET I
                                        ;*****************************************************************
001720' 042777  000002  001640  2S:     BIC     #2,@DLRCSR      ;TURN OFF DTR
001726' 005777  001634                  TST     @DLRCSR         ;DATA SET INT SHOULD HAVE SET
001732' 100407                          BMI     3$              ;BR IF IT DID
001734' 004767  001426                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001740' 005067  176142                  CLR     ERRTYP
                                        ;*****************************************************************
001744' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;DATA SET INT FAILED TO SET ON CLEARING DTR
                                        ;*****************************************************************
001752' 005777  001610          3S:     TST     @DLRCSR         ;ALL BITS NOW CLEAR ??
001756' 001407                          BEQ     DLT11           ;BR IF YES
```

```
001760' 004767  001402                  JSR     PC,STATR        ;GO SET UP ERROR INFO
001764' 005067  176116                  CLR     ERRTYP          ;
                                        ;*****************************************************************
001770' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;CLEARING DTR FAILED TO CLEAR OTHER BITS
                                        ;*****************************************************************


                                ;TEST THAT DATA SET INTR ENAB CAN SET AND CLEAR
                                ;----------------------------------------------

001776' 052777  000040  001562  DLT11:  BIS     #40,@DLRCSR     ;SET DATA SET I.E.
002004' 032777  000040  001554          BIT     #40,@DLRCSR     ;DID IT SET ??
002012' 001010                          BNE     1$              ;BR IF YES
002014' 004767  001346                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002020' 012767  000027  176060          MOV     #27,ERRTYP      ;
                                        ;*****************************************************************
002026' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;CAN'T SET DATA SET INTR.
                                        ;*****************************************************************
002034' 042777  000040  001524  1$:     BIC     #40,@DLRCSR     ;CLEAR DATA SET I.E.
002042' 005777  001520                  TST     @DLRCSR         ;DID IT CLEAR ??
002046' 001411                          BEQ     DLT12           ;BR IF YES
002050' 004767  001312                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002054' 012767  000027  176024          MOV     #27,ERRTYP      ;
                                        ;*****************************************************************
002062' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;CAN'T CLEAR DATA SET I.E.
                                        ;*****************************************************************
002070' 000463                          BR      DLT13           ;SKIP NEXT TEST


                                ;TEST THAT DATA SET INT CAN CAUSE A RCVR INTERRUPT WHEN ENABLED
                                ;--------------------------------------------------------------


002072' 005067  001512  DLT12:  CLR     INTFLG          ;INIT SOFTWARE INTR. FLAG
002076' 016700  175706          MOV     VECTOR,R0       ;GET RAS VECTOR ADDR.
002102' 012720  002174'         MOV     #2$,(R0)+       ;GO TO 2$ ON DATA SET INTERRUPT
002106' 005001                  CLR     R1              ;INIT BREAK TIMER
002110' 052777  000040  001450          BIS     #40,@DLRCSR     ;ENABLE DATA SET INTR.
002116' 052777  000010  001442          BIS     #10,@DLRCSR     ;SET SEC XMIT
002124'                         1$:
002124' 104407  000000'                 BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002130' 104407  000000'                 BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002134' 005767  001450                  TST     INTFLG          ;DID INTR OCCUR YET ??
002140' 001023                          BNE     3$              ;BR IF YES
002142' 005301                  2$:     DEC     R1              ;COUNT THE BREAK TIMER
002144' 001367                          BNE     1$              ;BR IF NO TIMEOUT
002146' 004767  001214                  JSR     PC,STATR        ;GO SET UP ERROR INFO
002152' 005077  001410                  CLR     @DLRCSR         ;TURN OFF THE I.E. BIT
002156' 012767  000023  175722          MOV     #23,ERRTYP      ;
                                        ;*****************************************************************
002164' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;DATA SET INTR. FAILED TO OCCUR
                                        ;*****************************************************************
002172' 000422                          BR      DLT13           ;GO TO NEXT TEST
002174' 042777  000040  001364  2$:     BIC     #40,@DLRCSR     ;TURN OFF DATA SET I.E.
002202' 005167  001402                  COM     INTFLG          ;SET SOFTWARE INTR. FLAG
002206' 000002                          RTI                     ;RETURN CONTROL TO OTHER GUY
002210' 032777  000040  001350  3$:     BIT     #40,@DLRCSR     ;DID INTR SERVICE TURN OFF I.E. ?
002216' 001410                          BEQ     DLT13           ;BR IF YES
002220' 004767  001142                  JSR     PC,STATR        ;GO SET UP ERROR INFO
```



```
002224' 012767  000023  175654          MOV     #23,ERRTYP      ;
                                        ;*****************************************************************
002232' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;INTR SERVICE FAILED TO CLR DATA SET I.E.
                                        ;*****************************************************************


                                ;TEST THAT "BREAK" BIT CAN SET AND CLEAR
                                ;---------------------------------------

002240' 052777  000001  001324  DLT13:  BIS     #1,@DLXCSR      ;SET BREAK BIT
002246' 032777  000001  001316          BIT     #1,@DLXCSR      ;DID IT SET ??
002254' 001010                          BNE     1$              ;BR IF YES
002256' 004767  001122                  JSR     PC,STATX        ;GO SET UP ERROR INFO
002262' 012767  000025  175616          MOV     #25,ERRTYP      ;
                                        ;*****************************************************************
002270' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;CAN'T SET BREAK BIT
                                        ;*****************************************************************
002276' 042777  000001  001266  1$:     BIC     #1,@DLXCSR      ;CLEAR THE BREAK BIT
002304' 032777  000001  001260          BIT     #1,@DLXCSR      ;DID IT CLEAR ?
002312' 001410                          BEQ     RFSTRT          ;BR IF YES
002314' 004767  001064                  JSR     PC,STATX        ;GO SET UP ERROR INFO
002320' 012767  000025  175560          MOV     #25,ERRTYP      ;
                                        ;*****************************************************************
002326' 104405  000000' 000000          HRDERR,BEGIN,NULL       ;BREAK BIT WON'T CLEAR
                                        ;*****************************************************************
```

```
                                ;               **********************
                                ;               *  SECTION TWO       *
                                ;               **********************
                                ;
                                RESTRT:
002334'                         CLR     XEND            ;CLEAR END FLAGS
002334' 005067  001242          CLR     @DLXCSR         ;CLEAR THE DL11 CONTROL REGS
002340' 005077  001226          CLR     @DLRCSR         ;JUST IN CASE
002344' 005077  001216          MOV     VECTOR,R0       ;GET START VECTOR ADDRESS
002350' 016700  175434          MOV     #RINT,(R0)+     ;SET UP THE RCVR AND XMIT VECTORS
002354' 012720  002422'         MOVB    #R1,(R0)
002360' 116710  175426          TST     (R0)+
002364' 005720                  MOV     #XINT,(R0)+
002366' 012720  002452'         MOVB    #R1,(R0)
002372' 116710  175414
002376' 012703  003612'         MOV     #LDTAB,R3       ;POINT TO TABLE OF LOAD SUBR. POINTERS
002402' 012704  003622'         MOV     #MTAB,R4        ;POINT TO TABLE OF MESSAGE POINTERS
002406' 005067  001174          CLR     RTRY            ;CLEAR RETRY FLAGS
002412' 005777  001152          TST     @DLRDBR         ;FLUSH RCVR INPUT BUFFER REG
002416' 005777  001146          TST     @DLRDBR
002422' 012367  001204          MOV     (R3)+,LDOUT     ;SET UP CORRECT LOAD BUF ADDRESS POINTER
002426' 012467  001202          MOV     (R4)+,AMESS     ;SET UP MESSAGE POINTER
002432' 004767  000436          JSR     PC,SEGX         ;GO DO A SEGMENT
002436' 022703  003622'         CMP     #MTAB,R3        ;DONE ALL FOUR SEGMENTS ??
002442' 001361                  BNE     1S              ;BR IF NOT
                                ;
                                ;
                                ;
                                ;
                                ;
002444' 104413  000000'         ENDITS,BEGIN            ;SIGNAL END OF ITERATION.
                                                        ;MONITOR SHALL TEST END OF PASS
002450' 000752                  BR      DOAGIN
                                ;
                                ;
```



```
                                ;THIS ROUTINE SERVICES ALL XMITTR INTRPTS. FOR ALL 256. BYTE XFERS
                                ;-----------------------------------------------------------------
002452' 105777  001114          XINT:   TSTB    @DLXCSR         ;XMIT READY SET ??
002456' 100403                  BMI     1S              ;BR IF YES
                                ;-----------------------------------------------------------------
002460' 000004  000000' 002516' PIRQS,BEGIN,4S           ; QUEUE UP TO CONTINUE AT 4S AND RTI
                                ;-----------------------------------------------------------------
002466' 022767  004522' 001104  1S:     CMP     #DLXBUFI,OPTR   ;OUTPUT 256. BYTES YET ??
002474' 001405                  BEQ     3S              ;BR IF YES
002476' 117777  001076  001070  2S:     MOVB    @OPTR,@DLXDBR   ;OUTPUT A CHARACTER
002504' 005267  001070          INC     OPTR            ;POINT TO NEXT CHAR. IN BUFFER
002510'                         3S:
                                ;-----------------------------------------------------------------
002510' 000004  000000' 002610' PIRQS,BEGIN,6S           ; QUEUE UP TO CONTINUE AT 6S AND RTI
                                ;-----------------------------------------------------------------
002516' 105767  001061          4S:     TSTB    XEND+1          ;ANY FATAL RCVR. ERRORS PENDING ??
002522' 001025                  BNE     5S              ;BR IF YES - STOP XMITTING
002524' 016767  001042  175146  MOV     DLXCSR,CSRA     ;SAVE THE CSR ADDRESS
002532' 017767  001034  175342  MOV     @DLXCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
002540' 042777  000104  001024  BIC     #104,@DLXCSR    ;DISABLE XMITTR INTERRUPTS
002546' 105167  001030          COMB    XEND            ;SET XMIT END FLAG
                                ;
002552' 012767  000011  175326  MOV     #11,ERRTYP
                                ;*****************************************************************
002560' 104405  000000' 000000  ERRORS,BEGIN,NULL       ;XMITTER FALSE INTERRUPT - FATAL ERROR
                                ;*****************************************************************
                                ;
002566' 005077  000774          CLR     @DLRCSR         ;TURN OFF RCVR INTR.
002572' 104410  000000'         ENDS,BEGIN              ;
                                ;
002576' 042777  000100  000766  5S:     BIC     #100,@DLXCSR    ;DISABLE XMITTR. INTERRUPTS
002604' 104400  000000'         EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
002610' 105767  000767          6S:     TSTB    XEND+1          ;ANY FATAL RCVR. ERRORS PENDING ??
002614' 001370                  BNE     6S              ;BR IF YES
002616' 104400  000000'         EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
                                ;
                                ;THIS ROUTINE SERVICES RECEIVER INTERRUPTS FOR ALL 256. BYTE XFERS
                                ;-----------------------------------------------------------------
                                ;
002622' 105777  000740          RINT:   TSTB    @DLRCSR         ;RCVR DONE SET ??
002626' 100403                  BMI     1S              ;BR IF YES
                                ;-----------------------------------------------------------------
002630' 000004  000000' 002702' PIRQS,BEGIN,3S           ; QUEUE UP TO CONTINUE AT 3S AND RTI
                                ;-----------------------------------------------------------------
002636' 005777  000726          1S:     TST     @DLRDBR         ;OVERRUN/PARITY/FRAMING ERRORS ??
002642' 100003                  BPL     2S              ;BR IF NONE
                                ;-----------------------------------------------------------------
002644' 000004  000000' 002774' PIRQS,BEGIN,5S           ; QUEUE UP TO CONTINUE AT 5S AND RTI
                                ;-----------------------------------------------------------------
002652' 022767  005122' 000716  2S:     CMP     #BUFEND,IPTR    ;INPUT BUFFER FULL ??
002660' 001405                  BEQ     7S              ;BR IF YES
002662' 117777  000702  000706  MOVB    @DLRDBR,@IPTR   ;READ THE DL INPUT BUFFER REG.
002670' 005267  000702          INC     IPTR            ;POINT TO NEXT CHAR. POSITION
```

```
002674'                          7$:
                                         ;-------------------------------------------------------------
002674' 000004  000000' 003062'          PIRQS,BEGIN,6$          ; QUEUE UP TO CONTINUE AT 6$ AND RTI
                                         ;-------------------------------------------------------------

002702' 105767  000674           3$:     TSTB    XEND            ;ANY FATAL XMITTR ERROR PENDING
002706' 001025                           BNE     4$              ;BR IF YES
002710' 016767  000652  175162           MOV     DLRCSR,CSRA     ;SAVE THE RCVR. CSR ADDRESS
002716' 017767  000644  175156           MOV     @DLRCSR,ACSR    ;SAVE CONTENTS OF CSR
002724' 042777  000100  000634           BIC     #100,@DLRCSR    ;TURN OFF THE RCVR.
002732' 105167  000645                   COMB    XEND+1          ;SET FATAL RCVR ERROR FLAG

002736' 012767  000011  175142           MOV     #11,ERRTYP
                                         ;*************************************************************
002744' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;RECEIVER FALSE INTERRUPT - FATAL ERROR
                                         ;*************************************************************

002752' 005077  000614                   CLR     @DLXCSR         ;DISABLE XMITTR TOO
002756' 104410  000000'                  ENDS,BEGIN              ;

002762' 042777  000100  000576   4$:     BIC     #100,@DLRCSR    ;DISABLE RCVR INTERRUPTS
002770' 104400  000000'                  EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

002774' 105767  000602           5$:     TSTB    XEND            ;ANY FATAL XMITTR ERRORS PENDING ??
003000' 001370                           BNE     4$              ;BR IF YES
003002' 016767  000560  175070           MOV     DLRCSR,CSRA     ;SAVE CSR ADDRESS
003010' 017767  000552  175064           MOV     @DLRCSR,ACSR    ;SAVE CONTENTS OF CSR
003016' 017767  000546  175060           MOV     @DLRDBR,ASTAT   ;SAVE THE ERROR FLAGS
003024' 042777  000100  000534           BIC     #100,@DLRCSR    ;DISABLE RCVR INTR.

003032' 012767  000017  175046           MOV     #17,ERRTYP
                                         ;*************************************************************
003040' 104405  000000' 000000           HRDERS,BEGIN,NULL       ;OVERRUN - PARITY - FRAMING ERROR
                                         ;*************************************************************

003046' 005077  000520                   CLR     @DLXCSR         ;DISABLE XMITTR TOO
003052' 105267  000530                   INCB    RTRY            ;SET RETRY FLAG
003056' 104400  000000'                  EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.

003062' 105767  000514           6$:     TSTB    XEND            ;ANY FATAL XMITTR ERRORS PENDING ??
003066' 001335                           BNE     4$              ;BR IF YES
003070' 104400  000000'                  EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
```



```
                                 ;THIS ROUTINE CONTROLS THE EXECUTION OF EACH OF THE FOUR DATA PATTERNS

003074' 000240                   SEGX:   NOP                     ;DO NOTHING FOR NOW
003076' 004767  000320           1$:     JSR     PC,CLDLBF       ;GO CLEAR BUFFERS
003102' 004777  000524                   JSR     PC,@LDOUT       ;GO SET UP  PATTERN
003106' 004767  000106                   JSR     PC,KICKOF       ;GO KICK OFF XMITTR AND RCVR
003112' 005002                           CLR     R2              ;INITIALIZE BREAK TIMER
003114'                          2$:
003114' 104407  000000'                  BREAKS,BEGIN            ;TEMPORARY RETURN TO MONITOR....
003120' 104407  000000'                  BREAKS,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
003124' 005767  000452                   TST     XEND            ;ANY FATAL ERRORS PENDING ??
003130' 001402                           BEQ     3$              ;BR IF NOT
003132' 104400  000000'                  EXITS,BEGIN                     ;EXIT TO MONITOR. MODULE WAIT FOR INTERRUPT.
003136' 105767  000444           3$:     TSTB    RTRY            ;RETRY FLAG SET ??
003142' 001411                           BEQ     4$              ;BR IF NOT
003144' 105067  000436                   CLRB    RTRY            ;CLEAR THE FLAG
003150' 105267  000433                   INCB    RTRY+1          ;COUNT ONE RETRY
003154' 122767  000003  000425           CMPB    #3,RTRY+1       ;TRIED THREE TIMES ??
003162' 001345                           BNE     1$              ;BR IF NOT - TRY IT AGAIN
003164' 000406                           BR      5$              ;REPORT IT AND GO TO NEXT SEGMENT
003166' 022767  005122' 000402   4$:     CMP     #BUFEND,IPTR    ;RECEIVED 256. CHARS. ??
003174' 001406                           BEQ     7$              ;BR IF YES
003176' 005302                           DEC     R2              ;DECREMENT BREAK COUNTER
003200' 001345                           BNE     2$              ;BR IF NO TIMEOUT
003202'                          5$:
003202' 104403  000000' 003634'          MSGNS,BEGIN,AMESS       ;ASCII MESSAGE CALL WITH COMMON HEADER
003210' 000207                   6$:     RTS     PC              ;GO TO NEXT SEGMENT
003212' 004767  000034           7$:     JSR     PC,CHKDAT       ;GO COMPARE IN/OUT DATA
003216' 000207                           RTS     PC              ;GO TO NEXT SEGMENT

                                 ;THIS ROUTINE KICKS OFF ALL 256. BYTE TRANSFERS
                                 ;----------------------------------------------

003220' 012767  004122' 000352   KICKOF: MOV     #DLBUFO,OPTR    ;POINT TO BEGINNING OF OUTPUT BUFFER
003226' 012767  004522' 000342           MOV     #DLBUFI,IPTR    ;POINT TO BEGINNING OF INPUT BUFFER
003234' 052777  000104  000330           BIS     #104,@DLXCSR    ;TURN ON XMITTER
003242' 052777  000100  000316           BIS     #100,@DLRCSR    ;TURN ON RCVR.
003250' 000207                           RTS     PC              ;RETURN TO CALLING SEGMENT

                                 ;THIS ROUTINE CHECKS FOR AND REPORTS DATA COMPARE ERRORS
                                 ;------------------------------------------------------

003252' 042777  000100  000312   CHKDAT: BIC     #100,@DLXCSR    ;DISABLE XMITTR INTR.
003260' 042777  000100  000300           BIC     #100,@DLRCSR    ;DISABLE RCVR INTR.
003266' 012700  004122'                  MOV     #DLBUFO,R0      ;R0 POINTS TO OUTPUT BUFFER
003272' 012701  004522'                  MOV     #DLBUFI,R1      ;R1 POINTS TO INPUT BUFFER
003276' 122021                   1$:     CMPB    (R0)+,(R1)+     ;COMPARE INPUT WITH OUTPUT
003300' 001004                           BNE     2$              ;BR IF NOT EQUAL
003302' 022701  005122'          3$:     CMP     #BUFEND,R1      ;END OF THE BUFFERS ??
003306' 001373                           BNE     1$              ;BR IF NOT
003310' 000207                           RTS     PC              ;RETURN TO CALLER

003312' 016767  000250  174560   2$:     MOV     DLRCSR,CSRA     ;SAVE THE CSR ADDRESS
003320' 114067  174562                   MOVB    -(R0),ASB       ;SAVE THE SHOULD BE DATA
003324' 042767  177400  174554           BIC     #177400,ASB     ;ZERO HI BYTE
003332' 010067  174544                   MOV     R0,SBADR        ;SAVE THE SHOULD BE ADDRESS
```

```
                                        MOVR    -(R1),AWAS      ;SAVE THE WAS DATA
                                        BIC     #177400,AWAS    ;CLEAR OUT HI BYTE
                                        MOV     R1,WASADR       ;SAVE THE WAS ADDRESS

                                        ;*****************************************************************
003354' 104404  000000'                 DATERR,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************

003360' 105720                          TSTB    (R0)+           ;POINT TO NEXT BYTE IN THE BUFFERS
003362' 105721                          TSTB    (R1)+
003364' 000746                          BR      3$              ;GO CHECK NEXT BYTE

                                ;THESE ROUTINES SET UP THE ERROR INFORMATION FOR THE BASIC TESTS
                                ;-------------------------------------------------------------

003366' 016767  000174  174504  STATR:  MOV     DLRCSR,CSRA     ;SAVE THE CSR ADDRESS
003374' 017767  000166  174500          MOV     @DLRCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
003402' 000207                          RTS     PC              ;RETURN TO BASIC TESTS

003404' 016767  000162  174466  STATX:  MOV     DLXCSR,CSRA     ;SAVE THE CSR ADDRESS
003412' 017767  000154  174462          MOV     @DLXCSR,ACSR    ;SAVE THE CONTENTS OF THE CSR
003420' 000207                          RTS     PC              ;RETURN TO THE BASIC TESTS

                                ;THIS ROUTINE IS USED TO CLEAR THE INPUT/OUTPUT BUFFERS
                                ;-----------------------------------------------------

003422' 012700  004122'         CLDLBF: MOV     #DLBUFO,R0      ;POINT R0 TO BEGINNING OF BUFFERS
003426' 005020                  1$:     CLR     (R0)+           ;CLEAR ONE WORD - UPDATE POINTER
003430' 022700  005122'                 CMP     #BUFEND,R0      ;DONE 256. BYTES ??
003434' 001374                          BNE     1$              ;BR IF NOT
003436' 000207                          RTS     PC              ;RETURN TO CALLING SEGMENT

                                ;THIS ROUTINE LOADS THE OUTPUT BUFFER WITH A NULL-DEL-NULL PATTERN
                                ;----------------------------------------------------------------

003440' 012700  004122'         LDOUT1: MOV     #DLBUFO,R0      ;SET UP POINTER
003444' 105020                  1$:     CLRB    (R0)+           ;MOV A NULL CHAR
003446' 112720  000377                  MOVB    #377,(R0)+      ;MOV A DELETE CHAR.
003452' 022700  004522'                 CMP     #DLBUFI,R0      ;BUFFER FULL ???
003456' 001372                          BNE     1$              ;BR IF NOT
003460' 000207                          RTS     PC              ;RETURN TO CALLING SEGMENT

                                ;THIS ROUTINE IS USED TO LOAD AN ASCENDING BINARY COUNT PATTERN
                                ;--------------------------------------------------------------

003462' 012700  004122'         LDOUT2: MOV     #DLBUFO,R0      ;SET UP POINTER
003466' 005001                          CLR     R1              ;USE R1 TO GENERATE THE PATTERN
003470' 110120                  1$:     MOVB    R1,(R0)+        ;LOAD ONE CHAR.
003472' 022700  004522'                 CMP     #DLBUFI,R0      ;BUFFER FULL ???
003476' 001402                          BEQ     2$              ;BR IF YES
003500' 105201                          INCB    R1              ;GENERATE NEXT CHAR.
003502' 000772                          BR      1$              ;GO MOVE IT
003504' 000207                  2$:     RTS     PC              ;RETURN TO CALLING SEGMENT

                                ;THIS ROUTINE IS USED TO LOAD THE DESCENDING BINARY COUNT PATTERN
```



```
                                ;-----------------------------------------------------------------

003506' 012700  004122'         LDOUT3: MOV     #DLBUFO,R0      ;SET UP POINTER
003512' 012701  000377                  MOV     #377,R1         ;START R1 AT 377
003516' 110120                  1$:     MOVB    R1,(R0)+        ;LOAD ONE CHAR.
003520' 022700  004522'                 CMP     #DLBUFI,R0      ;AT END OF THE BUFFER ??
003524' 001402                          BEQ     2$              ;BR IF YES
003526' 105301                          DECB    R1              ;GENERATE NEXT CHAR.
003530' 000772                          BR      1$              ;GO MOVE IT
003532' 000207                  2$:     RTS     PC              ;RETURN TO CALLING SEGMENT

                                ;THIS ROUTINE LOADS THE WORST CASE PATTERN
                                ;-----------------------------------------

003534' 012700  004122'         LDOUT4: MOV     #DLBUFO,R0      ;SET UP POINTERS
003540' 012701  005124'         1$:     MOV     #WCASE,R1       ;POINT TO MONITOR'S WORST CASE PATTERN
003544' 012120                  2$:     MOV     (R1)+,(R0)+     ;LOAD ONE WORD
003546' 022700  004522'                 CMP     #DLBUFI,R0      ;BUFFER FULL ???
003552' 001404                          BEQ     3$              ;BR IF YES
003554' 022701  005224'                 CMP     #WCASEE,R1      ;END OF WORST CASE PATTERN ??
003560' 001767                          BEQ     1$              ;GO RESET R1
003562' 000770                          BR      2$
003564' 000207                  3$:     RTS     PC              ;RETURN TO CALLING SEGMENT
```

```
                        ;VARIABLES,FLAGS,MESSAGES,AND BUFFERS
                        ;------------------------------------

003566' 000000          DLRCSR: OPEN            ;CONTAINS ADDRESS OF RCVR CSR
003570' 000000          DLRDBR: OPEN            ;CONTAINS ADDRESS OF RCVR DBR
003572' 000000          DLXCSR: OPEN            ;CONTAINS ADDRESS OF XMITTR CSR
003574' 000000          DLXDBR: OPEN            ;CONTAINS ADDRESS OF XMITTR DBR

003576' 000000          IPTR:   OPEN            ;CONTAINS POINTER TO INPUT BUFFER
003600' 000000          OPTR:   OPEN            ;CONTAINS POINTER TO OUTPUT BUFFER

003602' 000000          XFND:   OPEN            ;FATAL ERROR END FLAGS
003604' 000000          EPCTR:  OPEN            ;END OF PASS COUNTER
003606' 000000          RTRY:   OPEN            ;RETRY FLAG AND COUNTER
003610' 000000          INTFLG: OPEN                    ;SOFTWARE INTR. FLAG USED BY BASIC TESTS

003612' 003440'         LDTAB:  LDOUT1          ;POINTER TO 1ST LOAD BUFFER SUBR.
003614' 003462'                 LDOUT2          ;POINTER TO 2ND LOAD BUFFER ROUTINE
003616' 003506'                 LDOUT3          ;POINTER TO 3RD LOAD BUFFER ROUTINE
003620' 003534'                 LDOUT4          ;POINTER TO 4TH LOAD BUFFER ROUTINE
003622' 003644'         MTAB:   MSG1            ;POINTER TO MESSAGE 1
003624' 003705'                 MSG2            ;POINTER TO MESSAGE 2
003626' 003750'                 MSG3            ;POINTER TO MESSAGE 3
003630' 004015'                 MSG4            ;POINTER TO MESSAGE 4

003632' 000000          LDOUT:  OPEN            ;CONTAINS POINTER TO LOAD BUFFER SUBR.
003634' 003644'         AMESS:  MSG1            ;MESSAGE POINTERS
003636' 177777                  -1                      ;TERMINATOR
003640' 004053'         DRPMS:  MSG5                    ;MESSAGE POINTER
003642' 177777                  -1                      ;TERMINATOR

003644' 047045  046125  026514  MSG1:   .ASCIZ  /%NULL-DEL-NULL SEQUENCE ABORTED%/
003652' 042504  026514  052516
003660' 046114  051440  050505
003666' 042525  041516  020105
003674' 041101  051117  042524
003702' 022504      000
003705'     045 044502  040516  MSG2:   .ASCIZ  /%BINARY UP COUNT SEQUENCE ABORTED%/
003712' 054522  052440  020120
003720' 047503  047125  020124
003726' 042523  052521  047105
003734' 042503  040440  047502
003742' 052122  042105  000045
003750' 041045  047111  051101  MSG3:   .ASCIZ  /%BINARY DOWN COUNT SEQUENCE ABORTED%/
003756' 020131  047504  047127
003764' 041440  052517  052116
003772' 051440  050505  042525
004000' 041516  020105  041101
004006' 051117  042524  022504
004014'     000
004015'     045 047527  051522  MSG4:   .ASCIZ  /%WORST CASE SEQUENCE ABORTED%/
004022' 020124  040503  042523
004030' 051440  050505  042525
004036' 041516  020105  041101
004044' 051117  042524  022504
004052'     000
004053'     045 040506  040524  MSG5:   .ASCIZ  /%FATAL ERROR IN STATIC REGISTER TESTS%/
```



```
004060' 020114  051105  047522
004066' 020122  047111  051440
004074' 040524  044524  020103
004102' 042522  044507  052123
004110' 051105  052040  051505
004116' 051524  000045

                        .EVEN

                        ;512 WORDS RESERVED FOR TWO 256. BYTE BUFFERS
                        ;--------------------------------------------

004122' 000400          DLBUFO: .BLKB   256.    ;RSVD FOR OUTPUT BUFFER
004522' 000400          DLBUFI: .BLKB   256.    ;RSVD FOR INPUT BUFFER
005122' 000000          BUFEND: 0               ;MARK END OF BUFFER AREA

005124' 177776  000001  177775  WCASE:  .WORD   177776,1,177775,2,177773,4,177767,10
005132' 000002  177773  000004
005140' 177767  000010
005144' 177757  000020  177737          .WORD   177757,20,177737,40,177677,100,177577,200
005152' 000040  177677  000100
005160' 177577  000200
005164' 177377  000400  176777          .WORD   177377,400,176777,1000,175777,2000,173777,4000
005172' 001000  175777  002000
005200' 173777  004000
005204' 167777  010000  157777          .WORD   167777,10000,157777,20000,137777,40000,77777,100000
005212' 020000  137777  040000
005220' 077777  100000
005224'                 WCASE8:

            000001              .END
```

```
ACSR    000102R   389#   582*   824*   868*   896*   974*   978*
ADDR    000006R   355#   409
ADDR22= 001000    407#
AMESS   003634R   786*   927    1063#
ASR     000106R   393#   583*   955*   956*
ASTAT   000104R   391#   847*
AWAS    000110R   394#   563*   579    958*   959*
BEGIN   000000R   352#   437    439    440    450    452    453    464    466    467    474    476    477
                  490    491    499    501    502    511    513    514    529    531    532    539    541
                  542    548    549    559    561    562    575    577    578    587    607    614    622
                  634    641    649    656    669    676    684    691    703    711    725    726    735
                  746    758    766    798    810    818    830    834    837    841    849    854    862
                  874    878    881    892    897    901    910    911    914    927    963
BIT0  = 000001    407#
BIT1  = 000002    407#
BIT10 = 002000    407#
BIT11 = 004000    407#
BIT12 = 010000    407#
BIT13 = 020000    407#
BIT14 = 040000    407#
BIT15 = 100000    407#
BIT2  = 000004    407#
BIT3  = 000010    407#
BIT4  = 000020    407#
BIT5  = 000040    407#
BIT6  = 000100    407#
BIT7  = 000200    407#
BIT8  = 000400    407#
BIT9  = 001000    407#
BREAKS= 104407    407#   490    491    548    549    725    726    910    911
BR1     000012R   357#   486    522    778    781
BR2     000013R   358#
BTODS = 104421    407#
BUFEND  005122R   856    922    950    986    1106#
CDATAS= 104412    407#
CHKDAT  003252R   929    944#
CLDLBF  003422R   905    944#
CONFIG  000056R   377#
CSRA    000100R   387#   581*   823*   867*   885*   954*   973*   977*
DATCKS= 104411    407#
DATERS= 104404    407#   587    963
DLBUFF  004522R   812    936    947    996    1007   1019   1031   1105#
DLBUFO  004122R   935    946    984    993    1004   1016   1028   1104#
DLINIT  000256R   423#   410*   423*   424*   432    523*   524    533*   534    543*   555*   564*   568    571*
DLRCSR  003566R   581    582    601*   602    609    610*   617    626*   629    636    643*   644    651
                  662*   663*   664    671    678*   679    686    697*   698    705*   706    722*   723*
                  732*   738*   741    775*   833*   846    867    868    869*   880*   885    886    888*
                  938*   945*   954    973    974    1040#  785    786    851    858    887    1041#
DLRDBP  003570R   412*   426    427    563    584    785    786    851    858    887    1041#
DLT01   000302R   432#
DLT02   000342R   433    445#
DLT03   000404R   446    458#
DLT04   000524R   470    482#
DLT05   000712R   507    519#
DLT06   001344R   595    601#
```



```
DLT07   001466R   618    628#
DLT10   001634R   652    652#
DLT11   001776R   687    697#
DLT12   002072R   707    718#
DLT13   002240R   713    717    742    752#
DLXCSR  003572R   414*   425*   445    458*   459    468*   469    488*   503*   506    544*   556*   565*
                  572*   752*   753    760*   761    774*   807    823    824    825*   836*   877*   895*
                  937*   944*   977    978    1042#
DLXDBR  003574R   416*   546*   585    814*   1043#
DOACIN  002376R   772#   800
DRPMS   003640R   452    466    476    501    513    531    541    561    577    1065#
DVIDI   000014R   359#
ENDITS= 104413    407#   798
ENDS  = 104410    407#   440    453    467    477    502    514    532    542    562    578    834    878
EPCTR   003604R   1049#
ERRTYP  000106R   392#   435*   448*   462*   472*   497*   509*   527*   537*   557*   573*   605*   612*
                  620*   632*   639*   647*   654*   667*   674*   682*   689*   701*   709*   733*   744*
                  756*   764*   828*   872*   890*
EXITS = 104400    407#   817    841    891    897    901    914
GETPAS= 104415    407#
GWBUFS= 104414    407#
HRDCNT  000044R   372#
HRDERS= 104405    407#   437    450    464    474    499    511    529    539    559    575    607    614
                  622    634    641    649    656    669    676    684    691    703    711    735    746
                  758    766    830    874    892
HRDPAS  000050R   374#
ICONT   000036R   369#
ICOUNT  000040R   370#
IDNUM   000122R   399#
INIT    000030R   366#
INTPLC  003610R   482*   492    504*   519*   550    566*   718*   727    739*   1051#
INTP    000120R   398#
IPTR    003576R   856    858*   859*   922    936*   1045#
KICKOF  003220R   907    935#
LDOUT   003632R   787*   906    1062#
LDOUT1  003440R   993#   1053
LDOUT2  003462R   1004#  1054
LDOUT3  003506R   1016#  1055
LDOUT4  003514R   1028#  1056
LDTAB   003612R   782    1053#
MAP22S= 104416    407#
MODEM   001332R   580    594#
MODNAM  000000R   353#
MODSP   000224R   367    405#
MSG4S = 104403    407#   439    452    466    476    501    513    531    541    561    577    927
MSG5S = 104402    407#
MSGS  = 104401    407#
MSG1    003644R   1057   1063   1067#
MSG2    003705R   1058   1073#
MSG3    003750R   1059   1079#
MSG4    004015R   1060   1086#
MSG5    004053R   1065   1092#
MTAR    003622R   783    790    1057#
NULL  = 000000    407#   437    450    464    474    499    511    529    539    559    575    607    614
                  622    634    641    649    656    669    676    684    691    703    711    735    746
                  758    766    830    874    892
```

```
OPEN  = 000000            354     360     361     362     363     380     381     382     383     384     385     386     387
                          389     391     393     394     396     397     398    407#    1040    1041    1042    1043    1045
                         1046    1048    1049    1050    1051    1062
OPTR    003600R           812     814     815*    935*   1046#
OTOAS = 104420            407#
PASCNT  000034R           368#
PIRQS = 000004            407#    810     818     849     854     862
POPSP = 005726            407#
POPSP2= 022626            407#
PRTY  = 000000            358     407#
PRTY0 = 000000            407#
PRTY1 = 000040            407#
PRTY2 = 000100            407#
PRTY3 = 000140            407#
PRTY4 = 000200            357     407#
PRTY5 = 000240            407#
PRTY6 = 000300            407#
PRTY7 = 000340            407#
PS    = 177776            407#
PSW   = 177776            407#
PUSH  = 005746            407#
PUSH2 = 024646            407#
RANDS = 104417            407#
RANNUM  000054R           376#
RESTRT  002334R           395     596     762     772#
RES1    000056R           378#
RES2    000060R           379#
RINT    002622R           777     846#
RSTRT   000112R           395#
RTRY    003606R           784*    896*    915     917*    918*    919    1050#
SRADR   000102R           388#    585*    957*
SREGI   003074R           789     904#
SOFCNT  000042R           371#
SOFERS= 104406            407#
SOFPAS  000046R           373#
SPOINT  000032R           367#
SPSIZ = 000040              1#    400
SR1     000016R           360#    594
SR2     000020R           361#
SR3     000022R           362#
SR4     000024R           363#
START   000224R           366     409#
STAT    000026R           365#
STATR   003366R           434     526     536     554     570     604     611     619     631     638     646     653     666
                          673     681     688     700     708     731     743    973#
STATX   003404R           447     461     471     496     508     755     763    977#
SVR0    000062R           380#
SVR1    000064R           381#
SVR2    000066R           382#
SVR3    000070R           383#
SVR4    000072R           384#
SVR5    000074R           385#
SVR6    000076R           386#
SVSCNT  000052R           375#
TRPDFD= 000022            407#
VECTOR  000010R           356#    483     520     719     776
```



```
#ASADR  000104R           390#    584*    960*
WCASE   005124R          1029    1109#
WCASEF  005224R          1033    1121#
WDER    000116R           397#
#DTO    000114R           396#
XEND    003602R           773*    821     826*    839     865     870*    883     899     912    1048#
XFLAG   000005R           354#
XINT    002452R           780     807#
.       = 005224R          1104#   1105#

. ABS.  000000      000
        005224      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDLRB0,XDLRB0/SOL/CRF:SYM=DDXCOM,XDLRB0
 RUN-TIME:  2  3.3 SECONDS
 RUN-TIME RATIO: 31/6=4.9
 CORE USED:  7K  (13 PAGES)
```

DIAGNOSTIC ENGINEERING

digital **DECO☐ DEPO☒ SUBMISSION☐**

☐ NEW FOR RELEASE ENG. USE ☐ CHANGE ☐ DELETE

PRODUCT IDENTIFICATION

LIBRARY	PRODUCT NUMBER	REV	PATCH	ECO TALLY	PRODUCT DATE DD MMM YY	STATUS	DISTRIBUTION	1ST COPY-RIGHT YEAR	LAST COPY-RIGHT YEAR
ZZ	CXDLB	B	1	01	22 jan 79	☐ OBSOLETE	XX G ☐ R	1976	1979

TITLE CXDLBB1 DL11-E MODULE

AUTHOR D.BUTENHOF | MAINTAIN DEC/GROUP SPT GRP | MAINTAINER D.BUTENHOF | SUBMITTING ENGINEER D.BUTENHOF

PRODUCT COMPONENTS

CK	DESCRIPTION	PRODUCT NO.	REV	CK	DESCRIPTION	PRODUCT NO.	REV
	DOCUMENT				INDEX		
	LISTING				SOURCE MEDIA		
	OBJECT MEDIA				TEST MEDIA		
X	DEPO	AF-E998B-M1					

PRODUCTS OBSOLETED (other than previous version)

	LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV		LIBRARY	PRODUCT NUMBER	REV
MD				M D				M D			

PRODUCT CHARACTERISTICS

PROCESSORS PRODUCT OPERATES WITH (Enter all applicable 2-digit codes representing the Processor the product operates with. See separate instructions.)

03 04 05 10 20 21 34 35 40 45 50 55 60 70

OPERATIONAL CODES (Enter all applicable 2-digit codes that describe the product. See separate instructions.)

02 03 04 06 50

ACT/APT/XXDP INFORMATION FIELD	EXT	ACT SEQ NUMBER	ACT/XXDP COMPATIBLE?	APT COMPATIBLE?	1ST PASS RUN TIME	SUBSEQUENT PASS RUN TIME
			☒ Y ☐ N	☒ Y ☐ N	40 SECONDS	40 SECONDS

DECO/DEPO INFORMATION

PROBLEM REPORTS CLOSED:

DEVICE AFFECTED DEC/X11 | MULTIMEDIA AFFECTED? ☒ YES ☐ NO

KIT NUMBERS							
	ZJ129-RZ	ZJ129-FR	ZJ13Ø-RB				

PROBLEM:

Module is intented for 2Ø4Ø FRONT-END interface only, but documentation does not say so.

SOLUTION:

State in module header that this module is not intended for use on standalone PDP11 SYSTEMS.

DEPO PATCH AREA

CHANGE LOC	FROM	TO	CHANGE LOC	FROM	TO

SUBMITTING ENGINEER	MANUFACTURING ENGINEER	SUPPORT ENGINEER	CHARGE DECO/DEPO TO DISCRETE PROJECT NUMBER
DATE: 23-jan-79	DATE: 1-Feb-79	DATE:	Q99-05460
MAINTAIN ER	FIELD SERVICE	WAIVERING MANAGER	COORDINATION NO. MCF2838
23-jan-79	DATE:	DATE:	

7.0 MODULE OPERATION

BASIC TEST SEQUENCE:

DT01: VERIFY THAT ALL ZEROES CAN BE WRITTEN AND READ FROM THE "DELAY COUNTER" REGISTER

DT02: VERIFY THAT ALL ZEROES CAN BE WRITTEN AND READ FROM ALL ACTIVE "RAM" LOCATIONS

DT03: VERIFY THAT ALL ONES CAN BE WRITTEN AND READ FROM THE "DELAY COUNTER" REGISTER

DT04: VERIFY THAT ALL ONES CAN BE WRITTEN AND READ FROM ALL ACTIVE "RAM" LOCATIONS

DT05: VERIFY THAT EACH "RAM" LOCATION IS UNIQUELY ADDRESSABLE

DT06: VERIFY THAT THE "RMF=0" BIT DOES NOT SET WHEN A FLOATING ONE IS READ OUT OF THE "RAM"

DT07: VERIFY THAT THE "TO10 DONE" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT10: VERFIY THAT THE "10 REQ INT" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT11: VERIFY THAT THE "TO11 DONE" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT12: VERIFY THAT THE "TO10 ER" CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT13: VERIFY THAT THE "TO11 ER" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT14: VERIFY THAT "MSTR CLR" CAN CLEAR THE "TO11 BC" REG.

DT15: VERIFY THAT THE "ABC" REGISTER INCREMENTS DURING A "TO11 TRANSFER"

DT16: VERIFY THAT THE "TO11 DONE" BIT SETS PROPERLY

DT17: VERIFY THAT THE "STST NULL" FLOP CAN BE SET PROPERLY

DT20: VERIFY THAT THE "ABC" REGISTER INCREMENTS DURING A "TO10" E-BUFF FILL

8.0 OPERATOR OPTIONS

RELATIVE LOCATION "DTA 14" (DVID1) MUST BE MODIFIED TO EXERCISE MORE THAN ONE DTE20 AS SHOWN BELOW:

```
DVID1  BIT00=1  DTE20 #0        (DEFAULT)
DVID1  BIT01=1  DTE20 #1
DVID1  BIT02=1  DTE20 #2
DVID1  BIT03=1  DTE20 #3
```

TO DESELECT A DTE20 THE APPROPRIATE BIT IN "DVID1" MUST BE SET TO A ZERO. IF THE PROGRAM FINDS ALL FOUR BITS = "0" THE MODULE WILL BE DROPPED.

9.0 NON STANDARD PRINTOUTS

DTA USES THE DATA ERROR PRINTOUT IN SOME CASES TO REPORT OTHER THAN NORMAL DATA ERRORS. REFER TO THE ACTUAL ERROR CALL "APC" TO LOCATE THE CALL IN THE LISTING AND ANALYZE THE INSTRUCTIONS PRECEDING THE "DATER$" CALL TO OBTAIN THE INTERPRETION OF THE INFORMATION PRINTED.

.REM _

IDENTIFICATION

PRODUCT CODE: AC-E995D-MC

PRODUCT NAME: CXDTAD0 DTE20 MODULE

PRODUCT DATE: JANUARY 1981

MAINTAINER: DEC/X11 SUPPORT GROUP

1.0 ABSTRACT

"DTA" IS AN IOMOD THAT WILL EXERCISE UP TO FOUR DTE20'S SEQUENTIALLY. IT USES THE DIAGNOSTIC MODE TO VERIFY THE DTE20 UNIBUS INTERFACE AND THE LOGIC CONTROLLING THIS INTERFACE. IT PERFORMS SIMPLE RAM READ/WRITE AND ADDRESSING TESTS ALONG WITH VERIFICATION OF THE VECTORED INTERRUPT AND "NPR" FUNCTIONS. IT MAKES NO ATTEMPT TO COMMUNICATE WITH THE KL10 SIDE OF THE DEVICE.

2.0 REQUIREMENTS

HARDWARE: A PDP11 COMPUTER SYSTEM WITH AT LEAST ONE DTE20 KL10 INTERFACE.

STORAGE:: DTA REQUIRES:
1. DECIMAL WORDS: 852
2. OCTAL WORDS: 1524
3. OCTAL BYTES: 3250

3.0 PASS DEFINITION

THE FIRST PASS OF "DXDTA" CONSISTS OF EXECUTING EACH SUB-TEST ONE TIME. SUBSEQUENT PASSES CONSIST OF 100(8) ITERATIONS OF THE TEST SEQUENCE FOR EACH DTE20 FOUND.

4.0 EXECUTION TIME

PASS TIME WILL VARY DEPENDING UPON THE NO. OF DTE20'S SELECTED AND THE CONFIGURATION BEING EXERCISED.

5.0 CONFIGURATION PARAMETERS

DEFAULT PARAMETERS:

DVA: 174400 VCT: 774 BR1: 4 BR2: 0 DVC: 1

REQUIRED PARAMETERS:

TO EXERCISE MORE THAN ONE DTE20 "DVC" MUST BE SET UP AS DESCRIBED IN PARA. 8.0 BELOW.

6.0 DEVICE OPTION SET-UP

NONE REQUIRED

7.0 MODULE OPERATION

BASIC TEST SEQUENCE:

DT01: VERIFY THAT ALL ZEROES CAN BE WRITTEN AND READ FROM THE "DELAY COUNTER" REGISTER

DT02: VERIFY THAT ALL ZEROES CAN BE WRITTEN AND READ FROM ALL ACTIVE "RAM" LOCATIONS

DT03: VERIFY THAT ALL ONES CAN BE WRITTEN AND READ FROM THE "DELAY COUNTER" REGISTER

DT04: VERIFY THAT ALL ONES CAN BE WRITTEN AND READ FROM ALL ACTIVE "RAM" LOCATIONS

DT05: VERIFY THAT EACH "RAM" LOCATION IS UNIQUELY ADDRESSABLE

DT06: VERIFY THAT THE "RMF=0" BIT DOES NOT SET WHEN A FLOATING ONE IS READ OUT OF THE "RAM"

DT07: VERIFY THAT THE "TO10 DONE" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT10: VERFIY THAT THE "10 REQ INT" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT11: VERIFY THAT THE "TO11 DONE" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT12: VERIFY THAT THE "TO10 ER" CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT13: VERIFY THAT THE "TO11 ER" BIT CAN CAUSE AN INTERRUPT TO THE PROPER VECTOR

DT14: VERIFY THAT "MSTR CLR" CAN CLEAR THE "TO11 BC" REG.

DT15: VERIFY THAT THE "ABC" REGISTER INCREMENTS DURING A "TO11 TRANSFER"

DT16: VERIFY THAT THE "TO11 DONE" BIT SETS PROPERLY

DT17: VERIFY THAT THE "STST NULL" FLOP CAN BE SET PROPERLY

DT20: VERIFY THAT THE "ABC" REGISTER INCREMENTS DURING A "TO10" E-BUFF FILL

8.0 OPERATOR OPTIONS

RELATIVE LOCATION "DTA 14" (DVID1) MUST BE MODIFIED TO EXERCISE MORE THAN ONE DTE20 AS SHOWN BELOW:

```
DVID1  BIT00=1  DTE20 #0        (DEFAULT)
DVID1  BIT01=1  DTE20 #1
DVID1  BIT02=1  DTE20 #2
DVID1  BIT03=1  DTE20 #3
```

TO DESELECT A DTE20 THE APPROPRIATE BIT IN "DVID1" MUST BE SET TO A ZERO. IF THE PROGRAM FINDS ALL FOUR BITS = "0" THE MODULE WILL BE DROPPED.

9.0 NON STANDARD PRINTOUTS

DTA USES THE DATA ERROR PRINTOUT IN SOME CASES TO REPORT OTHER THAN NORMAL DATA ERRORS. REFER TO THE ACTUAL ERROR CALL "APC" TO LOCATE THE CALL IN THE LISTING AND ANALYZE THE INSTRUCTIONS PRECEDING THE "DATER$" CALL TO OBTAIN THE INTERPRETION OF THE INFORMATION PRINTED.

```
000000' IOMOD   <DTAD>,174400,774,4,0,0,4,157
000000'         MODULE  140000,DTAD,174400,774,4,0,0,4,157
                .TITLE  DTAD DEC/X11 SYSTEM EXERCISER MODULE
        ;       DDXCOM  VERSION  6      23-MAY-78
                        .LIST   BIN
                        ;*****************************************************************************
000000'                 BEGIN:
000000' 052104  042101      040 MODNAM: .ASCII  /DTAD / ;MODULE NAME.
000005'    000                  XFLAG:  .BYTE   OPEN            ;USED TO KEEP TRACK OF WBUFF USAGE
000006' 174400                  ADDR:   174400+0                ;1ST DEVICE ADDR.
000010' 000774                  VECTOR: 774+0                   ;1ST DEVICE VECTOR.
000012'    200                  BR1:    .BYTE   PRTY4+0         ;1ST BR LEVEL.
000013'    000                  BR2:    .BYTE   PRTY0+0         ;2ND BR LEVEL.
000014' 000001                  DVID1:  0+1                     ;DEVICE INDICATOR 1.
000016' 000000                  SR1:    OPEN                    ;SWITCH REGISTER 1
000020' 000000                  SR2:    OPEN                    ;SWITCH REGISTER 2
000022' 000000                  SR3:    OPEN                    ;SWITCH REGISTER 3
000024' 000000                  SR4:    OPEN                    ;SWITCH REGISTER 4
                                ;*****************************************************************************
000026' 140000                  STAT:   140000                  ;STATUS WORD.
000030' 000224'                 INIT:   START                   ;MODULE START ADDR.
000032' 000224'                 SPOINT: MODSP                   ;MODULE STACK POINTER.
000034' 000000                  PASCNT: 0                       ;PASS COUNTER.
000036' 000004                  ICONT:  4                       ;# OF ITERATIONS PER PASS=4
000040' 000000                  ICOUNT: 0                       ;LOC TO COUNT ITERATIONS
000042' 000000                  SOFCNT: 0                       ;LOC TO SAVE TOTAL SOFT ERRORS
000044' 000000                  HRDCNT: 0                       ;LOC TO SAVE TOTAL HARD ERRORS
000046' 000000                  SOFPAS: 0                       ;LOC TO SAVE SOFT ERRORS PER PASS
000050' 000000                  HRDPAS: 0                       ;LOC TO SAVE HARD ERRORS PER PASS
000052' 000000                  SYSCNT: 0                       ;# OF SYS ERRORS ACCUMULATED
000054' 000000                  RANNUM: 0                       ;HOLDS RANDOM # WHEN RAND MACRO IS CALLED
000056'                         CONFIG:                         ;RESERVED FOR MONITOR USE
000056' 000000                  RES1:   0                       ;RESERVED FOR MONITOR USE
000060' 000000                  RES2:   0                       ;RESERVED FOR MONITOR USE
000062' 000000                  SVR0:   OPEN                    ;LOC TO SAVE R0.
000064' 000000                  SVR1:   OPEN                    ;LOC TO SAVE R1.
000066' 000000                  SVR2:   OPEN                    ;LOC TO SAVE R2.
000070' 000000                  SVR3:   OPEN                    ;LOC TO SAVE R3.
000072' 000000                  SVR4:   OPEN                    ;LOC TO SAVE R4.
000074' 000000                  SVR5:   OPEN                    ;LOC TO SAVE R5.
000076' 000000                  SVR6:   OPEN                    ;LOC TO SAVE R6.
000100' 000000                  CSRA:   OPEN                    ;ADDR OF CURRENT CSR.
000102'                         SBADR:                          ;ADDR OF GOOD DATA, OR
000102' 000000                  ACSR:   OPEN                    ;CONTENTS OF CSR.
000104'                         WASADR:                         ;ADDR OF BAD DATA, OR
000104' 000000                  ASTAT:  OPEN                    ;STATUS REG CONTENTS.
000106'                         ERRTYP:                         ;TYPE OF ERROR
000106' 000000                  ASB:    OPEN                    ;EXPECTED DATA.
000110' 000000                  AWAS:   OPEN                    ;ACTUAL DATA.
000112' 000246'                 RSTRT:  RESTRT                  ;RESTART ADDRESS AFTER END OF PASS
000114' 000000                  WDTO:   OPEN                    ;WORDS TO MEMORY PER ITERATION
000116' 000000                  WDFR:   OPEN                    ;WORDS FROM MEMORY PER ITERATION
000120' 000000                  INTR:   OPEN                    ;# OF INTERRUPTS PER ITERATION
000122' 000157                  IDNUM:  157                     ;MODULE IDENTIFICATION NUMBER=157
        000040                          .REPT   SPSIZ           ;MODULE STACK STARTS HERE.
```



```
                                .NLIST
                                .WORD   0
                                .LIST
                                .ENDR
000224'                         MODSP:
                                ;*****************************************************************************
239
240
241             000040          DTESIZ= 000040                  ;16 REGISTERS PER DTE20
242             000004          DTEMAX= 000004                  ;UP TO FOUR DTE'S CAN BE TESTED
243             020000          TO11BM= BIT13                   ;TO-11 BYTE MODE
244             002000          DS05=   BIT10                   ;DIAGNOSTIC STATUS
245             000060          PULSE=  BIT4!BIT5               ;SINGLE PULSE THE 10/11 CLOCK
246             000040          D1011=  BIT5                    ;10/11 INTERFACE DIAGNOSTIC MODE
247             000200          TO10=   BIT7                    ;INTERFACE MAJOR STATE = TO10 TRANSFER
248             040000          EDONES= BIT14                   ;SET E-BUS DONE
249             000100          DRESET= BIT6                    ;PERFORM DIAGNOSTIC CLEAR
250             000040          INTRON= BIT5                    ;ENABLE DTE20 TO INTR. THE 11
251             000002          ERR11S= BIT1                    ;SET TO11 ERROR
252             100000          TO10DN= BIT15                   ;TO10 DONE
253             010000          RAMIS0= BIT12                   ;OUTPUT READ FROM RAM IS ALL ZEROES
254             000400          TO10DB= BIT8                    ;REQ 10 INTERRUPT = DOORBELL FROM 11
255             000040          NULSTP= BIT5                    ;NULL STOP
256             020000          ERR10S= BIT13                   ;SET TO 10 ERROR
257             100000          DON10S= BIT15                   ;SET TO 10 DONE
258             000200          DON11S= BIT7                    ;SET TO 11 DONE
259             000100          TO11=   BIT6                    ;INTERFACE MAJOR STATE = TO11 XFR
260             004000          TO11DB= BIT11
261             000200          TO11DN= BIT7
262             040000          ZSTOP=  BIT14           ;STOP ON NULL (ZERO) CHAR
263
264 000224' 016767  177564  003000  START:  MOV     DVID1,TDVD1             ;GET DEVICE SELECT PARAMETER
265 000232' 042767  177760  002772          BIC     #177760,TDVD1           ;CLEAR OUT UNUSED BITS
266 000240' 001002                          BNE     RESTRT                  ;BR IF ANY DTE20'S SELECTED
267 000242' 104410  000000'                 END$,BEGIN              ;
268
269 000246' 016767  002760  002760  RESTRT: MOV     TDVD1,TDVD2             ;SAVE THE SELECT BITS
270 000254' 016705  177526                  MOV     ADDR,R5                 ;GET THE FIRST DTE20 ADDRESS
271 000260' 016700  177524                  MOV     VECTOR,R0               ;GET THE FIRST VECTOR ADDRESS
272 000264' 000257                  1$:     CCC                             ;CLEAR OUT THE "C" BIT
273 000266' 006067  002742                  ROR     TDVD2                   ;"C" WILL SET IF A DTE IS SELECTED
274 000272' 103002                          BCC     2$                      ;BR IF SELECT BIT = 0
275 000274' 004767  000016                  JSR     PC,GOEXDT               ;GO EXERCISE SELECTED DTE
276 000300' 062705  000040          2$:     ADD     #DTESIZ,R5              ;GENERATE NEXT DTE START ADDR.
277 000304' 162700  000004                  SUB     #DTEMAX,R0              ;GENERATE NEXT DTE VECTOR ADDR.
278 000310' 104413  000000'                 ENDIT$,BEGIN                    ;SIGNAL END OF ITERATION.
279                                                                         ;MONITOR SHALL TEST END OF PASS
280 000314' 000763                          BR      1$
281
282 000316' 012702  003172'         GOEXDT: MOV     #DLYCNT,R2              ;POINT TO FIRST TABLE ENTRY
283 000322' 005003                          CLR     R3                      ;INIT R3 TO COUNT BY +2
284 000324' 010512                  1$:     MOV     R5,(R2)                 ;STORE A DTE ADDRESS
285 000326' 060322                          ADD     R3,(R2)+                ;MAKE IT THE RIGHT ADDRESS
286 000330' 005723                          TST     (R3)+                   ;ADD +2 TO R3
287 000332' 022703  000040                  CMP     #DTESIZ,R3              ;STORED ALL DTE ADDRESSES ??
288 000336' 001372                          BNE     1$                      ;BR IF NOT
```

```
000340' 016767  002676  002670          MOV     ITCNT,ICOUN             ;INITIALIZE ITERATION COUNTER
000346' 005367  002664          DOAGIN: DEC     ICOUN                   ;COUNT ONE ITERATION

                                ;ZEROES TEST TO DELAY COUNTER
                                ;----------------------------

000352' 005067  177530          DT01:   CLR     ASB                     ;RESULT S/B = 000000
000356' 004767  002356                  JSR     PC,DIAGRT               ;GO DO DIAGNOSTIC RESET
000362' 004767  002366                  JSR     PC,DIAGME               ;GO SET UP DIAGNOSTIC MODE
000366' 012777  000000  002576          MOV     #0,@DLYCNT              ;ZERO THE DELAY COUNT REG.
000374' 017767  002572  177506          MOV     @DLYCNT,AWAS            ;GET THE DELAY COUNT REG.
000402' 001412                          BEQ     DT02                    ;BR IF DATA CORRECT
000404' 010567  177470                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
000410' 016767  002556  177466          MOV     DLYCNT,WASADR
000416' 012767  000106' 177456          MOV     #ASB,SBADR
                                        ;****************************************************************
000424' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;****************************************************************

                                ;TEST TO WRITE ZEROES IN ALL ACTIVE RAM LOCATIONS
                                ;-----------------------------------------------

000430' 005067  177452          DT02:   CLR     ASB                     ;RESULT S/B = 000000
000434' 016703  002532                  MOV     DLYCNT,R3               ;POINT R3 TO FIRST RAM LOC.
000440' 004767  002274          1$:     JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
000444' 004767  002304                  JSR     PC,DIAGME               ;GO SET IT IN DIAGNOSTIC MODE
000450' 005023                          CLR     (R3)+                   ;ZERO A RAM LOCATION
000452' 020367  002544                  CMP     R3,DIAG1                ;DONE ALL ACTIVE LOCATIONS ??
000456' 001370                          BNE     1$                      ;BR IF NOT
000460' 004767  002254          2$:     JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
000464' 004767  002264                  JSR     PC,DIAGME               ;GO SET DIAGNOSTIC MODE
000470' 014367  177414                  MOV     -(R3),AWAS              ;GET CONTENTS OF RAM
000474' 001411                          BEQ     3$                      ;BR IF IT WAS 000000
000476' 010367  177402                  MOV     R3,WASADR               ;SAVE THE ERROR INFO
000502' 012767  000106' 177372          MOV     #ASB,SBADR
000510' 010567  177364                  MOV     R5,CSRA
                                        ;****************************************************************
000514' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;****************************************************************
000520' 032777  010000  002500  3$:     BIT     #RAMISO,@STATUS         ;DID RMF=0 BIT SET
000526' 001012                          BNE     4$                      ;BR IF IT DID
000530' 010567  177344                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
000534' 017767  002466  177342          MOV     @STATUS,ASTAT
000542' 005067  177340                  CLR     ERRTYP
                                        ;****************************************************************
000546' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;RMF=0 FAILED TO SET
                                        ;****************************************************************
000554' 020367  002412          4$:     CMP     R3,DLYCNT               ;CHECKED ALL LOCATIONS ??
000560' 003337                          BGT     2$                      ;BR IF NOT

                                ;TEST TO WRITE ALL 1'S INTO DELAY COUNTER
                                ;----------------------------------------

000562' 012767  177777  177316  DT03:   MOV     #-1,ASB                 ;RESULT S/B = 177777
000570' 004767  002144                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
000574' 004767  002154                  JSR     PC,DIAGME               ;GO SET UP DIAGNOSTIC MODE
```



```
000600' 012777  177777  002364          MOV     #-1,@DLYCNT             ;ALL 1'S TO DELAY COUNT REG.
000606' 017767  002360  177274          MOV     @DLYCNT,AWAS            ;GET THE CONTENTS OF DELAY COUNT
000614' 026767  177266  177266          CMP     ASB,AWAS                ;WAS IT ALL ONES ??
000622' 001412                          BEQ     DT04                    ;BR IF YES
000624' 016767  002342  177252          MOV     DLYCNT,WASADR           ;SAVE THE ERROR INFO
000632' 012767  000106' 177242          MOV     #ASB,SBADR
000640' 010567  177234                  MOV     R5,CSRA
                                        ;****************************************************************
000644' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;****************************************************************

                                ;TEST TO WRITE ALL 1'S INTO ACTIVE RAM LOCATIONS
                                ;----------------------------------------------

000650' 016703  002316          DT04:   MOV     DLYCNT,R3               ;R3 POINTS TO 1ST ADDRESS
000654' 004767  002060          1$:     JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
000660' 004767  002070                  JSR     PC,DIAGME               ;GO SET DIAGNOSTIC MODE
000664' 012723  177777                  MOV     #-1,(R3)+               ;LOAD ALL 1'S INTO RAM
000670' 020367  002326                  CMP     R3,DIAG1                ;LOADED ALL LOCATIONS ??
000674' 001367                          BNE     1$                      ;BR IF NOT
000676' 012767  177777  177202  2$:     MOV     #-1,ASB                 ;RESULT S/B = 177777
000704' 004767  002030                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
000710' 004767  002040                  JSR     PC,DIAGME               ;GO SET DIAGNOSTIC MODE
000714' 014367  177170                  MOV     -(R3),AWAS              ;READ  A RAM LOCATION
000720' 026767  177162  177162          CMP     ASB,AWAS                ;WAS IT 177777 ??
000726' 001411                          BEQ     3$                      ;BR IF YES
000730' 010367  177150                  MOV     R3,WASADR               ;SAVE THE ERROR INFO
000734' 012767  000106' 177140          MOV     #ASB,SBADR
000742' 010567  177132                  MOV     R5,CSRA
                                        ;****************************************************************
000746' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;****************************************************************
000752' 020367  002214          3$:     CMP     R3,DLYCNT               ;CHECKED THEM ALL YET ??
000756' 003347                          BGT     2$                      ;BR IF NOT

                                ;RAM MEMORY ADDRESS TEST
                                ;----------------------

001000' 016703  002206          DT05:   MOV     DLYCNT,R3               ;START WITH 1ST RAM LOC
001004' 005004                          CLR     R4                      ;R4 CONTAINS FOUR ZEROBYTES (4 BITS EACH)
001006' 004767  001746                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001012' 004767  001756                  JSR     PC,DIAGME               ;GO SET DIAGNOSTIC MODE
001016' 010423                  1$:     MOV     R4,(R3)+                ;LOAD A RAM LOCATION-UPDATE POINTER
001020' 062704  021042                  ADD     #21042,R4               ;ADD +2 TO EACH DATA BYTE
001024' 020367  002212                  CMP     R3,DIAG1                ;LOADED THEM ALL ??
001030' 001372                          BNE     1$                      ;BR IF NOT
001032' 162704  021042          2$:     SUB     #21042,R4               ;INIT R4 TO START CHECKING DATA
001036' 010467  177064                  MOV     R4,ASB                  ;SAVE S/B DATA
001042' 016367  177776  177060          MOV     -2(R3),AWAS             ;GET CONTENTS OF RAM
001050' 020443                          CMP     R4,-(R3)                ;CORRECT CONTENTS ??
001052' 001411                          BEQ     3$                      ;BR IF YES
001054' 010367  177044                  MOV     R3,WASADR               ;SAVE THE ERROR DATA
001060' 012767  000106' 177034          MOV     #ASB,SBADR
001066' 010567  177026                  MOV     R5,CSRA
                                        ;****************************************************************
001072' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
```

```
                                        ;*****************************************************************
001056' 005704                  3$:     TST     R4                      ;BACK TO 000000 DATA
001060' 001354                          BNE     2$                      ;BR IF NOT

                                        ;TEST THAT RMF=0 DOES NOT SET WITH FLOATING ONE OUT OF RAM
                                        ;-------------------------------------------------------

001062' 004767  001652          DT06:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001066' 005067  177014                  CLR     ASB                     ;RESULT S/B = 000000
001072' 012703  000001                  MOV     #BIT0,R3                ;INIT R3 TO LOAD DLYCNT
001076' 010377  002070          1$:     MOV     R3,@DLYCNT              ;LOAD FLOATING ONE INTO DLYCNT
001102' 017767  002064  177000          MOV     @DLYCNT,AWAS            ;READ IT BACK OUT
001110' 032777  010000  002110          BIT     #RAMISO,@STATUS         ;DID RMF=0 GET SET ??
001116' 001415                          BEQ     2$                      ;BR IF NOT -- ITS OK
001120' 016767  176764  176754          MOV     AWAS,ACSR               ;[CSRC] = DATA READ
001126' 010567  176746                  MOV     R5,CSRA                 ;SAVE ERROR INFO
001132' 017767  002070  176744          MOV     @STATUS,ASTAT
001140' 005067  176742                  CLR     ERRTYP
                                        ;*****************************************************************
001144' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;RMF=0 BIT FAILED TO GET CLEARED
                                        ;*****************************************************************
001152' 000241                  2$:     CLC                             ;PUT 0'S IN ON RIGHT SIDE
001154' 006103                          ROL     R3                      ;ROTATE THE FLOATING ONE
001156' 001347                          BNE     1$                      ;BR TILL IT COMES OUT ON LEFT

                                        ;TEST THAT "TO10 DONE" CAUSES A VECTORED INTERRUPT
                                        ;-------------------------------------------------

001160' 004767  001554          DT07:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001164' 005067  002050                  CLR     INTFLG                  ;CLEAR SOFTWARE FLAG
001170' 012720  003164'                 MOV     #DTINT,(R0)+            ;GO TO DTINT ON INTERRUPT
001174' 116710  176612                  MOVB    BR1,(R0)
001200' 005740                          TST     -(R0)                   ;RESET VECTOR POINTER
001202' 012777  100040  002016          MOV     #DON10S!INTRON,@STATUS  ;ENABLE "TO10" TO CAUSE INTR.
001210' 000240                          NOP                             ;DELAY A LITTLE
001212' 000240                          NOP                             ;TO ALLOW INTERRUPT
001214' 000240                          NOP
001216' 005767  002016                  TST     INTFLG                  ;DID THE INTERRUPT OCCUR ??
001222' 001015                          BNE     DT10                    ;BR IF IT DID
001224' 010567  176650                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
001230' 012767  100040  176646          MOV     #DON10S!INTRON,ASTAT
001236' 004767  001476                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001242' 012767  000023  176636          MOV     #23,ERRTYP
                                        ;*****************************************************************
001250' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"TO10 DONE" FAILED TO CAUSE INTERRUPT
                                        ;*****************************************************************


                                        ;TEST THAT "10 REQ INT" CAN GENERATE VECTORED INTERRUPT
                                        ;-----------------------------------------------------

001256' 004767  001456          DT10:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001262' 005067  001752                  CLR     INTFLG                  ;CLEAR SOFTWARE FLAG
001266' 012777  004040  001732          MOV     #TO11DB!INTRON,@STATUS  ;ENABLE "10 REQ INT" TO CAUSE INTR
001274' 000240                          NOP                             ;A LITTLE STALL
001276' 000240                          NOP
001300' 000240                          NOP
```



```
001302' 005767  001732                  TST     INTFLG                  ;DID THE INTR. OCCUR ??
001306' 001015                          BNE     DT11                    ;BR IF YES
001310' 010567  176564                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
001314' 012767  004040  176562          MOV     #TO11DB!INTRON,ASTAT
001322' 004767  001412                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001326' 012767  000023  176552          MOV     #23,ERRTYP
                                        ;*****************************************************************
001334' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"10 REQ INT" FAILED TO GENERATE INTERRUPT
                                        ;*****************************************************************


                                        ;TEST THAT "TO11 DONE" CAN GENERATE A VECTORED INTERRUPT
                                        ;-------------------------------------------------------

001342' 004767  001372          DT11:   JSR     PC,DIAGRT               ;GO DO DIAGNOSTIC RESET
001346' 005067  001666                  CLR INTFLG                      ;CLEAR SOFTWARE INTR. FLAG
001352' 012777  000240  001646          MOV     #DON11S!INTRON,@STATUS  ;ENABLE "TO11 DONE" TO CAUSE INTR
001360' 000240                          NOP                             ;WAIT A LITTLE
001362' 000240                          NOP
001364' 000240                          NOP
001366' 005767  001646                  TST     INTFLG                  ;DID INTR. OCCUR ??
001372' 001015                          BNE     DT12                    ;BR IF IT DID
001374' 010567  176500                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
001400' 012767  000240  176476          MOV     #DON11S!INTRON,ASTAT    ;
001406' 004767  001326                  JSR     PC,DIAGRT               ;GO DO DIAGNOSTIC RESET
001412' 012767  000023  176466          MOV     #23,ERRTYP
                                        ;*****************************************************************
001420' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"TO11 DONE FAILED TO CAUSE INTERRUPT
                                        ;*****************************************************************


                                        ;TEST THAT "TO10 ER" CAN CAUSE A VECTORED INTERRUPT
                                        ;--------------------------------------------------

001426' 004767  001306          DT12:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001432' 005067  001602                  CLR     INTFLG                  ;INIT SOFTWARE FLAG
001436' 012777  020040  001562          MOV     #ERR10S!INTRON,@STATUS  ;ENABLE "TO10ER" TO CAUSE INTR.
001444' 000240                          NOP                             ;A LITTLE STALL
001446' 000240                          NOP
001450' 000240                          NOP
001452' 005767  001562                  TST     INTFLG                  ;DID INTR. OCCUR ??
001456' 001015                          BNE     DT13                    ;BR IF IT DID
001460' 010567  176414                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
001464' 012767  020040  176412          MOV     #ERR10S!INTRON,ASTAT
001472' 004767  001242                  JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001476' 012767  000023  176402          MOV     #23,ERRTYP
                                        ;*****************************************************************
001504' 104405  000000' 000000          HRDERS,BEGIN,NULL       ;"TO10ER" FAILED TO CAUSE AN INTR.
                                        ;*****************************************************************


                                        ;TEST THAT "TO11ER" CAN CAUSE A VECTORED INTERRUPT
                                        ;-------------------------------------------------

001512' 005067  001522          DT13:   CLR     INTFLG
001516' 012777  000042  001502          MOV     #ERR11S!INTRON,@STATUS  ;ENAB "TO11ER" TO CAUSE INTR.
001524' 000240                          NOP                             ;A LITTLE STALL
001526' 000240                          NOP
001530' 000240                          NOP
```

```
001532' 005767  001502          TST     INTFLG                  ;DID THE INTR OCCUR ??
001536' 001015                  BNE     1$                      ;BR IF IT DID
001540' 010567  176334          MOV     R5,CSRA                 ;SAVE THE ERROR INFO
001544' 012767  000042  176332  MOV     #ERR11S!INTRON,ASTAT
001552' 004767  001162          JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001556' 012767  000023  176322  MOV     #23,ERRTYP
                                ;*****************************************************************
001564' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;"TU11ER" FAILED TO CAUSE AN INTERRUPT
                                ;*****************************************************************
001572' 010010          1$:     MOV     R0,(R0)                 ;RESTORE TRAP CATCHER IN DTE
001574' 062720  000002          ADD     #2,(R0)+                ;BEFORE LEAVING
001600' 005010                  CLR     (R0)
001602' 005740                  TST     -(R0)

                        ;TEST TO VERIFY "MSTR CLR" CAN CLEAR "TO11 BC"
                        ;---------------------------------------------

001604' 004767  001130  DT14:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001610' 016767  001432  176270  MOV     NPRLOC,ASB              ;GET NPR ADDRESS
001616' 012777  007777  001364  MOV     #7777,@TO11BC           ;LOAD TO11 BYTE COUNT
001624' 004767  001110          JSR     PC,DIAGRT               ;GO DO A DIAGNOSTID RESET
001630' 016777  001412  001356  MOV     NPRLOC,@TO11AD          ;LOAD TO 11 ADDRESS
001636' 000240                  NOP                             ;STALL A LITTLE
001640' 000240                  NOP
001642' 000240                  NOP
001644' 017767  001344  176236  MOV     @TO11AD,AWAS            ;GET THE TO 11 ADDRESS
001652' 026767  176230  176230  CMP     ASB,AWAS                ;DID IT GET MODIFIED ??
001660' 001412                  BEQ     DT15                    ;BR IF NOT
001662' 016767  001326  176214  MOV     TO11AD,WASADR           ;SAVE THE ERROR INFO
001670' 012767  000106' 176204  MOV     #ASB,SBADR
001676' 010567  176176          MOV     R5,CSRA
                                ;*****************************************************************
001702' 104404  000000'         DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************

                        ;TEST TO VERIFY ABC REG INCREMENTS DURING TO11 TRANSFER
                        ;-----------------------------------------------------

001706' 016767  001334  176172  DT15:   MOV     NPRLOC,ASB              ;GET THE NPR ADDRESS
001714' 005267  176166          INC     ASB                     ;INCREMENT IT
001720' 004767  001014          JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
001724' 016703  001316          MOV     NPRLOC,R3               ;GET NPR ADDRESS AGAIN
001730' 062703  000002          ADD     #2,R3                   ;ADD +2 TO IT
001734' 004767  001034          JSR     PC,T11LOC               ;GO LOCK IN TO11 MAJOR STATE
001740' 005067  176142          CLR     ERRTYP
                                ;*****************************************************************
001744' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;FAILED TO LOCK IN "TO11" MAJOR STATE
                                ;*****************************************************************
001752' 000450                  BR      DT16                    ;GO TO NEXT TEST
001754' 012777  027777  001226  MOV     #TO11BM!7777,@TO11BC    ;LOAD TO 11 BYTE COUNT REG.
001762' 016777  001260  001224  MOV     NPRLOC,@TO11AD          ;LOAD TO 11 ADDR. REG.
001770' 012777  040010  001226  MOV     #10!EDONES,@DIAG2       ;GO SET UP DIAG REG.
001776' 004767  000722          JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002002' 012777  000014  001214  MOV     #14,@DIAG2              ;CHANGE DIAG REG.
002010' 004767  000710          JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002014' 012777  000020  001202  MOV     #20,@DIAG2              ;CHANGE THE DIAG REG.
```



```
002022' 004767  000676          JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002026' 017767  001162  176054  MOV     @TO11AD,AWAS            ;GET THE TO 11 ADDRESS
002034' 026767  176046  176046  CMP     ASB,AWAS                ;DID IT INCREMENT PROPERLY ??
002042' 001414                  BEQ     DT16                    ;BR IF IT DID
002044' 016767  001144  176032  MOV     TO11AD,WASADR           ;SAVE THE ERROR INFO
002052' 012767  000106' 176022  MOV     #ASB,SBADR
002060' 010567  176014          MOV     R5,CSRA
002064' 004767  000650          JSR     PC,DIAGRT               ;GO DO DIAGNOSTIC RESET
                                ;*****************************************************************
002070' 104404  000000'         DATER$,BEGIN            ;DATA ERROR!!!
                                ;*****************************************************************


                        ;TEST TO CHECK THE "TO11 DONE" FLAG GETS SET
                        ;-------------------------------------------

002074' 012767  000200  176004  DT16:   MOV     #TO11DN,ASB             ;TO11DN BIT SHOULD GET SET
002102' 004767  000632          JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
002106' 005077  001076          CLR     @TO11BC                 ;CLEAR TO 11 BYTE COUNT REG.
002112' 004767  000622          JSR     PC,DIAGRT               ;GO DO A DIAGRT
002116' 016777  001124  001070  MOV     NPRLOC,@TO11AD          ;LOAD THE TO 11 ADDR REG.
002124' 004767  000644          JSR     PC,T11LOC               ;GO LOCK IN TO 11 MAJOR STATE
002130' 005067  175752          CLR     ERRTYP
                                ;*****************************************************************
002134' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;FAILED TO LOCK IN TO 11 STATE
                                ;*****************************************************************
002142' 000504                  BR      DT17                    ;GO TO NEXT TEST
002144' 005077  001040          CLR     @TO11BC                 ;LOAD TO 11 BYTE COUNT REG.
002150' 005077  001050          CLR     @DIAG2                  ;START AT MINOR STATE "TO11 DLY RD"
002154' 005077  001042          CLR     @DIAG1                  ;TURN CLOCK BACK ON
002160' 005004                  CLR     R4                      ;INIT BREAK TIMER
002162'                 1$:
002162' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002166' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002172' 032777  000100  001022  BIT     #TO11,@DIAG1            ;DID WE LEAVE THE TO11 STATE ??
002200' 001415                  BEQ     2$                      ;BR IF YES
002202' 005304                  DEC     R4                      ;COUNT THE TIMER
002204' 001366                  BNE     1$                      ;BR IF NO TIME OUT
002206' 010567  175666          MOV     R5,CSRA                 ;SAVE THE ERROR INFO
002212' 017767  001004  175664  MOV     @DIAG1,ASTAT
002220' 005067  175662          CLR     ERRTYP
                                ;*****************************************************************
002224' 104405  000000' 000000  HRDER$,BEGIN,NULL       ;FAILED TO LEAVE THE TO11 STATE
                                ;*****************************************************************
002232' 000450                  BR      DT17                    ;GO TO NEXT TEST
002234' 005004          2$:     CLR     R4                      ;INIT THE TIMER AGAIN
002236'                 3$:
002236' 104407  000000'         BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
002242' 104407  000000'         BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
002246' 032777  000100  000746  BIT     #TO11,@DIAG1            ;BACK IN TO11 STATE YET
002254' 001015                  BNE     4$                      ;BR IF YES
002256' 005304                  DEC     R4                      ;COUNT THE BREAK TIMER
002260' 001366                  BNE     3$                      ;BR IF NO TIMEOUT
002262' 017767  000734  175614  MOV     @DIAG1,ASTAT            ;SAVE THE ERROR INFO
002270' 010567  175604          MOV     R5,CSRA
002274' 005067  175606          CLR     ERRTYP
```

```
                                        ;*****************************************************************
002300' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO GET BACK IN TO11 STATE
                                        ;*****************************************************************
002306' 000422                          BR      DT17                    ;GO TO NEXT TEST
002310' 000240                  4$:     NOP
002312' 017767  000710  175570          MOV     @STATUS,AWAS            ;GET THE STATUS REG.
002320' 032767  000200  175562          BIT     #TO11DN,AWAS            ;TO11 DONE BIT SET ?
002326' 001012                          BNE     DT17                    ;BR IF YES
002330' 010567  175544                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
002334' 012767  000110' 175542          MOV     #AWAS,WASADR
002342' 012767  000106' 175532          MOV     #ASB,SBADR
                                        ;*****************************************************************
002350' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************


                                ;CHECK THAT STST NULL FLOP CAN BE SET
                                ;------------------------------------

002354' 012767  000040  175524  DT17:   MOV     #NULSTP,ASB             ;NULSTP SHOULD GET SET
002362' 004767  000402                  JSR     PC,T11LOK               ;LOCK IN TO 11 STATE
002366' 005067  175514                  CLR     ERRTYP
                                        ;*****************************************************************
002372' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO LOCK IN TO 11 STATE
                                        ;*****************************************************************
002400' 000446                          BR      DT20                    ;GO TO NEXT TEST
002402' 012777  077777  000600          MOV     #ZSTOP!37777,@TO11BC    ;SET THE NULL STOP BIT
002410' 005077  000604                  CLR     @TO11DT                 ;CLEAR TO 11 DATA REG.
002414' 016777  000626  000572          MOV     NPRLOC,@TO11AD          ;LOAD THE TO11 ADDR REG.
002422' 012777  040010  000574          MOV     #10!EDONES,@DIAG2       ;LOAD DIAG2 REG
002430' 004767  000270                  JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002434' 012777  000014  000562          MOV     #14,@DIAG2              ;SET UP DIAG2
002442' 004767  000256                  JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002446' 017767  000554  175434          MOV     @STATUS,AWAS            ;GET THE STATUS REG.
002454' 042767  177737  175426          BIC     #^C<NULSTP>,AWAS        ;CLEAR OUT JUNK BITS
002462' 026767  175420  175420          CMP     ASB,AWAS                ;DID NULSTP SET ??
002470' 001412                          BEQ     DT20                    ;BR IF YES
002472' 010567  175402                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
002476' 012767  000110' 175400          MOV     #AWAS,WASADR
002504' 012767  000106' 175370          MOV     #ASB,SBADR
                                        ;*****************************************************************
002512' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************


                                ;CHECK ABC INCREMENTATION DURING TO 10 E-BUFF FILL
                                ;------------------------------------------------

002516' 004767  000216          DT20:   JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
002522' 016767  000520  175356          MOV     NPRLOC,ASB              ;SAVE NPR ADDRESS
002530' 062767  000001  175350          ADD     #1,ASB                  ;INCREMENT IT
002536' 016777  000504  000446          MOV     NPRLOC,@TO10AD          ;LOAD THE TO 10 ADDR REG.
002544' 004767  000322                  JSR     PC,T10LOC               ;GO LOCK IN TO 10 STATE
002550' 005067  175332                  CLR     ERRTYP
                                        ;*****************************************************************
002554' 104405  000000' 000000          HRDER$,BEGIN,NULL       ;FAILED TO LOCK IN TO 10 STATE
                                        ;*****************************************************************
002562' 000441                          BR      DTEXIT                  ;EXIT TESTS
```



```
002564' 012777  000010  000432          MOV     #10,@DIAG2              ;SET UP TO SHIFT STATES
002572' 004767  000126                  JSR     PC,DIAG9P               ;GO PULSE THE CLOCK
002576' 012777  040000  000420          MOV     #EDONES,@DIAG2          ;SET EDONES BIT
002604' 000240                          NOP
002606' 012777  000014  000410          MOV     #14,@DIAG2              ;SHIFT STATES
002614' 004767  000104                  JSR     PC,DIAG9P               ;PULSE THE CLOCK TWICE
002620' 004767  000100                  JSR     PC,DIAG9P
002624' 017767  000362  175256          MOV     @TO10AD,AWAS            ;GET THE TO 10 ADDRESS
002632' 026767  175250  175250          CMP     ASB,AWAS                ;DID IT GET INCREMENTED ??
002640' 001412                          BEQ     DTEXIT                  ;BR IF YES
002642' 010567  175232                  MOV     R5,CSRA                 ;SAVE THE ERROR INFO
002646' 016767  000340  175230          MOV     TO10AD,WASADR
002654' 012767  000106' 175220          MOV     #ASB,SBADR
                                        ;*****************************************************************
002662' 104404  000000'                 DATER$,BEGIN            ;DATA ERROR!!!
                                        ;*****************************************************************


002666' 005767  175142          DTEXIT: TST     PASCNT                  ;FIRST TIME THROUGH ?
002672' 001405                          BEQ     1$                      ;BR IF YES-QUICK PASS
002674' 005767  000336                  TST     ICOUN                   ;DO IT AGAIN ??
002700' 001402                          BEQ     1$                      ;BR IF NO
002702' 000167  175440                  JMP     DOAGIN                  ;GO DO IT
002706' 000207                  1$:     RTS     PC                      ;GO TO NEXT DTE20


                                ;COMMON SUBROUTINES
                                ;------------------

002710' 012777  000060  000304  DIAGPU: MOV     #PULSE!D1011,@DIAG1     ;PULSE THE CLOCK
002716' 000240                          NOP
002720' 000240                          NOP
002722' 000207                          RTS     PC

002724' 012777  002060  000270  DIAG9P: MOV     #DS05!PULSE!D1011,@DIAG1
002732' 000240                          NOP
002734' 000240                          NOP
002736' 000207                          RTS     PC

002740' 012777  000100  000256  DIAGRT: MOV     #DRESET,@DIAG2          ;DIAGNOSTIC RESET
002746' 000240                          NOP
002750' 000240                          NOP
002752' 000207                          RTS     PC

002754' 012777  000040  000240  DIAGME: MOV     #D1011,@DIAG1           ;SET DIAGNOSTIC MODE
002762' 000240                          NOP
002764' 000240                          NOP
002766' 000207                          RTS     PC

002770' 004767  177744          T11LOK: JSR     PC,DIAGRT
002774' 004767  177754          T11LOC: JSR     PC,DIAGME
003000' 005004                  T11LKA: CLR     R4                      ;INIT BREAK TIMER
003002' 032777  000100  000212  1$:     BIT     #TO11,@DIAG1            ;IN TO11 STATE YET ??
003010' 001017                          BNE     T11KB                   ;BR IF YES
003012' 004767  177672                  JSR     PC,DIAGPU               ;GO PULSE THE CLOCK
003016' 104407  000000'                 BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
```

```
  737  003022' 104407  000000'          BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
  738  003026' 005304                   DEC     R4                      ;COUN THE TIMER
  739  003030' 001364                   BNE     1$                      ;BR IF NO TIMEOUT
  740  003032' 016767  000164  175040    MOV     DIAG1,CSRA              ;SAVE THE ERROR INFO
  741  003040' 017767  000156  175034    MOV     @DIAG1,ACSR
  742  003046' 000207                   RTS     PC                      ;ERROR RETURN
  743  003050' 012777  002040  000144 T11KB: MOV  #D1011!DS05,@DIAG1      ;SET STATE HOLD
  744  003056' 062716  000012            ADD     #12,(SP)                ;MOVE PC AROUND ERROR CALL
  745  003062' 000240                   NOP
  746  003064' 000207                   RTS     PC                      ;NOW RETURN
  747
  748  003066' 004767  177646 T10LOK: JSR     PC,DIAGRT               ;GO DO A DIAGNOSTIC RESET
  749  003072' 004767  177656 T10LOC: JSR     PC,DIAGME               ;GO SET DIAG MODE
  750  003076' 005004                   CLR     R4                      ;INIT TIME OUT COUNTER
  751  003100'                       1$:
  752  003100' 104407  000000'          BREAK$,BEGIN            ;TEMPORARY RETURN TO MONITOR....
  753  003104' 104407  000000'          BREAK$,BEGIN            ;THEN CONTINUE AT NEXT INSTRUCTION.
  754  003110' 032777  000200  000104    BIT     #TO10,@DIAG1            ;SHIFT TO TO 10 STATE YET ?
  755  003116' 001013                   BNE     T10LKB                  ;BR IF YES
  756  003120' 004767  177564            JSR     PC,DIAGPU               ;PULSE THE CLOCK
  757  003124' 005304                   DEC     R4                      ;COUNT THE TIMER
  758  003126' 001364                   BNE     1$                      ;BR IF NO TIMEOUT
  759  003130' 016767  000066  174742    MOV     DIAG1,CSRA              ;SAVE THE ERROR INFO
  760  003136' 017767  000060  174736    MOV     @DIAG1,ACSR
  761  003144' 000207                   RTS     PC                      ;ERROR RETURN
  762  003146' 012777  002040  000046 T10LKB: MOV  #D1011!DS05,@DIAG1      ;LOCK IN TO 10 STATE
  763  003154' 062716  000012            ADD     #12,(SP)                ;MOVE PC AROUND ERROR CALL
  764  003160' 000240                   NOP
  765  003162' 000207                   RTS     PC                      ;NORMAL RETURN
  766
  767
  768  003164' 005167  000050 DTINT:  COM     INTFLG                  ;SET SOFTWARE FLAG
  769  003170' 000002                   RTI
  770
  771                                ;CONSTANTS,ADDRESS TABLE, AND VARIABLES
  772                                ;--------------------------------------
  773
  774                                ;THIS TABLE GETS LOADED WITH THE ADDRESSES OF THE DTE REGISTERS
  775
  776  003172' 000000                DLYCNT: 0
  777  003174' 000000                DEXWD3: 0
  778  003176' 000000                DEXWD2: 0
  779  003200' 000000                DEXWD1: 0
  780  003202' 000000                TENAD1: 0
  781  003204' 000000                TENAD2: 0
  782  003206' 000000                TO10BC: 0
  783  003210' 000000                TO11BC: 0
  784  003212' 000000                TO10AD: 0
  785  003214' 000000                TO11AD: 0
  786  003216' 000000                TO10DT: 0
  787  003220' 000000                TO11DT: 0
  788  003222' 000000                DIAG1:  0
  789  003224' 000000                DIAG2:  0
  790  003226' 000000                STATUS: 0
  791  003230' 000000                DIAG3:  0
  792
```



```
  793                                ;VARIABLES AND FLAGS
  794
  795  003232' 000000                TDVD1:  0                       ;DEVICE SELECT BITS BUFFERS
  796  003234' 000000                TDVD2:  0
  797  003236' 000000                ICOUN:  0                       ;PASS ITERATION COUNTER
  798  003240' 000000                INTFLG: 0                       ;SOFTWARE INTERRUPT FLAG
  799  003242' 000100                ITCNT:  100                     ;NO. OF TEST ITERATIONS FOR EA. DTE20
  800  003244' 000000                NPRBUF: 0                       ;NPR XFER BUFFER
  801
  802                                ;CONSTANTS
  803
  804  003246' 003244'                NPRLOC: NPRBUF                  ;ADDRESS POINTER TO NPR BUFFER
  805
  806          000001                .END
```

```
ACSR    000102R          221#    415*    741*    760*
ADDR    000006R          187#    270
ADDR22= 001000           239#
ASB     000106R          225#    295*    303     311*    323     342*    347     350     365*    369     372     392*    397
                         409*    531*    539     542     551*    552*    571     574     585*    635     643*    659     663
                         672*    673*    689     693
ASTAT   000104R          223#    331*    417*    441*    460*    479*    498*    516*    608*    622*
AWAS    000110R          226#    299*    320*    346*    347     368*    369     393*    412*    415     538*    539     570*
                         571     630*    631     634     657*    658*    659     662     688*    689
BEGIN   000000R          184#    267     278     305     326     334     353     375     400     420     445     464     483
                         502     520     545     559     578     593     601     602     611     616     617     626     637
                         647     665     678     695     736     737     752     753
BIT0  = 000001           239#    410
BIT1  = 000002           239#    251
BIT10 = 002000           239#    244
BIT11 = 004000           239#    260
BIT12 = 010000           239#    253
BIT13 = 020000           239#    243     256
BIT14 = 040000           239#    248     262
BIT15 = 100000           239#    252     257
BIT2  = 000004           239#
BIT3  = 000010           239#
BIT4  = 000020           239#    245
BIT5  = 000040           239#    245     246     250     255
BIT6  = 000100           239#    249     259
BIT7  = 000200           239#    247     258     261
BIT8  = 000400           239#    254
BIT9  = 001000           239#
BREAK$= 104407           239#    601     602     616     617     736     737     752     753
BR1     000012R          189#    432
BR2     000013R          190#
BTOD$ = 104421           239#
CDATA$= 104412           239#
CONFIG  000056R          209#
CSRA    000100R          219#    301*    324*    330*    351*    373*    398*    416*    440*    459*    478*    497*    515*
                         543*    575*    607*    623*    633*    661*    691*    740*    759*
DATCK$= 104411           239#
DATER$= 104404           239#    305     326     353     375     400     545     578     637     665     695
DEXWD1  003200R          779#
DEXWD2  003176R          778#
DEXWD3  003174R          777#
DIAGME  002754R          297     314     319     344     361     367     386     725#    731     749
DIAGPU  002710R          710#    735     756
DIAGRT  002740R          296     313     318     343     360     366     385     408     429     442     451     461     470
                         480     489     499     517     530     533     553     576     586     588     671     720#    730
                         748
DIAG1   003222R          316     363     389     598*    603     608     618     622     710*    715*    725*    733     740
                         741     743*    754     759     760     762*    788#
DIAG2   003224R          564*    566*    568*    597*    653*    655*    681*    683*    685*    720*    789#
DIAG3   003230R          791#
DIAG9P  002724R          565     567     569     654     656     682     686     687     715#
DLYCNT  003172R          282     298*    299     302     312     336     345*    346     349     359     377     383     411*
                         412     776#
DOAG1N  000346R          290#    703
DON10S= 100000           257#    434     441
DON11S= 000200           258#    472     479
```



```
DRESET= 000100           249#    720
DS05  = 002000           244#    715     743     762
DTEMAX= 000004           242#    277
DTESIZ= 000040           241#    276     287
DTEXIT  002666R          680     690     699#
DTINT   003164R          431     768#
DT01    000352R          295#
DT02    000430R          300     311#
DT03    000562R          342#
DT04    000650R          348     359#
DT05    000760R          383#
DT06    001062R          408#
DT07    001160R          429#
DT10    001256R          439     451#
DT11    001342R          458     470#
DT12    001426R          477     489#
DT13    001512R          496     508#
DT14    001604R          530#
DT15    001706R          540     551#
DT16    002074R          561     572     585#
DT17    002354R          595     613     628     632     643#
DT20    002516R          649     660     671#
DVID1   000014R          191#    264
D1011 = 000040           246#    710     715     725     743     762
EDONES= 040000           248#    564     653     683
ENDIT$= 104413           239#    278
END$  = 104410           239#    267
ERRTYP  000106R          224#    332*    418*    443*    462*    481*    500*    518*    557*    591*    609*    624*    645*
                         676#
ERR10S= 020000           256#    491     498
ERR11S= 000002           251#    509     516
EXIT$ = 104400           239#
GETPA$= 104415           239#
GOEXDT  000316R          275     282#
GWBUF$= 104414           239#
HRDCNT  000044R          204#
HRDER$= 104405           239#    334     420     445     464     483     502     520     559     593     611     626     647
                         678
HRDPAS  000050R          206#
ICONT   000036R          201#
ICOUN   003236R          289#    290*    701     797#
ICOUNT  000040R          202#
IDNUM   000122R          231#
INIT    000030R          198#
INTFLG  003240R          430#    438     452*    457     471*    476     490*    495     508*    513     768*    798#
INTR    000120R          230#
INTRON= 000040           250#    434     441     453     460     472     479     491     498     509     516
ITCNT   003242R          289     799#
MAP22$= 104416           239#
MODNAM  000000R          185#
MODSP   000224R          199     237#
MSGN$ = 104403           239#
MSGS$ = 104402           239#
MSG$  = 104401           239#
NPRBUF  003244R          800#    804
NPRLOC  003246R          531     534     551     554     563     589     652     672     674     804#
```

```
NULL  = 000000          239#  334   420   445   464   483   502   520   559   593   611   626   647
                        678
NULSTP= 000040          255#  643   658
OPEN  = 000000          186   192   193   194   195   212   213   214   215   216   217   218   219
                        221   223   225   226   228   229   230   239#
OTOA$ = 104420          239#
PASCNT  000034R         200#  699
PIRQ$ = 000004          239#
POPSP = 005726          239#
PUPSP2= 022626          239#
PRTY  = 000000          239#
PRTY0 = 000000          190   239#
PRTY1 = 000040          239#
PRTY2 = 000100          239#
PRTY3 = 000140          239#
PRTY4 = 000200          189   239#
PRTY5 = 000240          239#
PRTY6 = 000300          239#
PRTY7 = 000340          239#
PS    = 177776          239#
PSW   = 177776          239#
PULSE = 000060          245#  710   715
PUSH  = 005746          239#
PUSH2 = 024646          239#
RAMISO= 010000          253#  328   413
RAND$ = 104417          239#
RANNUM  000054R         208#
RESTRT  000246R         227   266   269#
RES1    000056R         210#
RES2    000060R         211#
RSTRT   000112R         227#
SBADR   000102R         220#  303*  323*  350*  372*  397*  542*  574*  635*  663*  693*
SOFCNT  000042R         203#
SOFERS= 104406          239#
SOFPAS  000046R         205#
SPOINT  000032R         199#
SPSIZ = 000040            1#  232
SR1     000016R         192#
SR2     000020R         193#
SR3     000022R         194#
SR4     000024R         195#
START   000224R         198   264#
STAT    000026R         197#
STATUS  003226R         328   331   413   417   434*  453*  472*  491*  509*  630   657   790#
SVR0    000062R         212#
SVR1    000064R         213#
SVR2    000066R         214#
SVR3    000070R         215#
SVR4    000072R         216#
SVR5    000074R         217#
SVR6    000076R         218#
SYSCNT  000052R         207#
TDVD1   003232R         264*  265*  269   795#
TDVD2   003234R         269*  273*  796#
TENAD1  003202R         780#
TENAD2  003204R         781#
```



```
TO10  = 000200          247#  754
TO10AD  003212R         674*  688   692   784#
TO10BC  003206R         782#
TO10DB= 000400          254#
TO10DN= 100000          252#
TO10DT  003216R         786#
TO11  = 000100          259#  603   618   733
TO11AD  003214R         534*  538   541   563*  570   573   589*  652*  785#
TO11BC  003210R         532*  562*  587*  596*  650*  783#
TO11BM= 020000          243#  562
TO11DB= 004000          260#  453   460
TO11DN= 000200          261#  585   631
TO11DT  003220R         651*  787#
TRPDFD= 000022          239#
T10LKB  003146R         755   762#
T10LOC  003072R         675   749#
T10LOK  003066R         748#
T11KB   003050R         734   743#
T11LKA  003000R         732#
T11LOC  002774R         556   590   731#
T11LOK  002770R         644   730#
VECTOR  000010R         188#  271
WASADR  000104R         222#  302*  322*  349*  371*  396*  541*  573*  634*  662*  692*
WDFR    000116R         229#
WDTO    000114R         228#
XFLAG   000005R         186#
ZSTOP = 040000          262#  650
```

```
BKMOD       1#
BREAK       1#    600     615     736     751
BTOD        1#
CKDATA      1#
DATACK      1#
DATERR      1#    304     325     352     374     399     544     577     636     664     694
DFSEVN      1#    239
DSEVNT      1#    239
END         1#    267
ENDIT       1#    278
ENDMOD      1#
EQUATS      1#    239
EXIT        1#
GETPA       1#
GWBUFF      1#
HRDER       1#    333     419     444     463     482     501     519     558     592     610     625     646     677
IOMOD       1#    178
IOMODP      1#
IOMODR      1#
IOMODX      1#
MAP22       1#
MODULE      1#    179
MSG         1#
MSGN        1#
MSGS        1#
NBKMOD      1#
OTOA        1#
PIRQ        1#
RAND        1#
SBKMOD      1#
SOFER       1#

. ABS.  000000      000
        003250      001

 ERRORS DETECTED:  0
 DEFAULT GLOBALS GENERATED:  0

 XDTAD0.OBJ,XDTAD0.LST/CRF/SOL/NL:TOC=DDXCOM.P11,XDTAD0.P11
 RUN-TIME: 4 7 1 SECONDS
 RUN-TIME RATIO: 40/13=2.9
 CORE USED:  7K  (13 PAGES)

 DOCUMENT PAGES:  21
```