PDT11/150

SYSTEM EXCERCISER

CVKDAA0

AH-E443A-MC

FICHE 1 OF 1

DEC 1978

digital

MADE IN USA

.REM %

IDENTIFICATION

PRODUCT CODE: AC-E442A-MC

PRODUCT NAME: CVKDAA0 PDT11/150 SYSTEM EXERCISER

PRODUCT DATE: SEPT 1978

MAINTAINER: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).

THE PDT-11/150 SYSTEM EXERCISER TESTS THE PROCESSORS ABILITY
TO OPERATE ALL ITS PERIPHERALS IN INTERRUPT MODE AT THE SAME TIME
WITH EMPHASIS ON THE DISK SUBSYSTEMS.
IT SHOULD BE NOTED THAT IT IS NOT A DIAGNOSTIC OF THE PERIPHERALS
BUT A SERIES OF SYSTEM INTERACTION TESTS.

THE PROGRAM IS DEFAULTED TO EXERCISE THE CLUSTER TERMINALS & THE
COMM PORT IN INTERNAL LOOPBACK (MAINT) MODE .
THE DEFAULT CAN BE CHANGED TO EXERCISE ANY OF THE CLUSTER TERMINALS
AND/OR THE COMM PORT IN EXTERNAL LOOPBACK.
SEE PROGRAM OPTIONS SEC. 2.4.

TESTING OF THE ACTUAL CLUSTER TERMINALS OR COMM DEVICE IS NOT PERFORMED.
THESE DEVICES CAN ONLY BE TESTED EITHER IN EXT. OR INT. LOOPBACK.

THE PRINTER LOGIC WILL BE EXERCISED IF SIZING DETERMINES IT TO BE PRESENT
OR IF DATA TERM RDY IS ASSERTED ON ITS CONNECTOR.

THE CONSOLE TERMINAL IS NOT TESTED.

AFTER THE MEMORY HAS BEEN TESTED & ALL THE PERIPHERALS ARE
BEING EXERCISED & INTERRUPTING AT RANDOM, THE DISK SUBSYSTEM TESTS WILL BEGIN.
SEE TEST SUMMARY SEC. 7.0.

THE PROGRAM CONTAINS A UTILITY WHICH CAN COPY THE SYSTEM EXERCISER DISK
FROM DX0 ONTO A SCRATCH DISK IN DX1.
THIS ENABLES THE OPERATOR TO CREATE BACKUP COPIES OF THE EXERCISER DISK.
SEE SECTION 2.8.

THE PROGRAM ALSO CONTAINS A COMPATABILITY TESTING OPTION.
SEE SEC. 2.7.

1.2 SYSTEM REQUIREMENTS.

HARDWARE REQUIREMENTS:

PDT-11/150 SYSTEM
8K MEMORY - MINIMUM
CONSOLE TERMINAL
EXTERNAL LOOPBACK CONNECTORS (OPTIONAL) FOR COMM & CLUSTER TERMINAL PORTS.
(SEE PROGRAM OPTIONS SEC. 2.4)
VT-100 CONSOLE MUST BE SETUP FOR JUMP SCROLL.

SOFTWARE REQUIREMENTS:

THIS EXERCISER IS DESIGNED TO RUN IN ANY OF THE FOLLOWING WAYS:

STAND ALONE
WITH APT MONITOR
WITH RT-11 MONITOR

IT IS NOT DESIGNED TO RUN WITH THE DIAGNOSTIC SUPERVISOR.

1.3 RELATED DOCUMENTS AND STANDARDS.

DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS	175-003-009-02
APT	MD-11-DZZMA
SYSMAC	MD-11-DZQAC

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

THIS EXERCISER ASSUMES THAT THE POWER UP SELF TEST RUNS ERROR FREE.

1.5 ASSUMPTIONS

THIS EXERCISER ASSUMES THAT THE OPERATOR HAS MODIFIED THE DEVICE MAP
($DEVM) AT LOC. 1246 IF THE DEFAULTS DO NOT AGREE WITH THE ACTUAL
SYSTEM CONFIGURATION. SEE PROGRAM OPTIONS SEC. 2.4.
THE PROGRAM ALSO ASSUMES THE SOFTWARE SWITCH REGISTER IS
PROPERLY SET UP (SWREG) AT LOC. 176. SEE SEC. 2.3.

1.6 RUNNING SYSTEMS PROGRAMS

UNLESS THE SYSTEMS PROGRAMS ARE KNOWN TO INITIALIZE THE BAUD
RATES OF EACH DEVICE THRU THE PARAMETER REGISTER,
THE OPERATOR SHOULD POWER DOWN & UP AGAIN WHEN FINISHED RUNNING THIS EXERCISER.
THIS IS TO RESTORE THE PDT-11/150 TO ITS DEFAULT BAUD RATES.
OTHERWISE, THE DEVICES WILL ATTEMPT TO RUN AT WHATEVER BAUD RATES
WERE LAST USED BY THIS EXERCISER.

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.

USE STANDARD PROCEDURE FOR PDP-11 ABSOLUTE BINARY FORMATTED PAPER TAPE.
THE PROGRAM WILL AUTO-START AFTER LOADING.

THE SYSTEM EXERCISER DISK WILL AUTO-START AFTER BOOTING.

EXAMPLE OF STARTUP ASSUMING PRINTER NOT PRESENT, 24K MEMORY,
& TERM #2 NOT TO BE TESTED, COMM PORT EXT. LOOPBACK, TERM 1 & 3 INT. LOOPBACK.
(20007 IN $DEVM, SEE SEC. 1.5):

```
        PDT-11/150 SYSTEM EXERCISER

        SWR = 000000    NEW = 110000 <CR>       (HALT ON ERR, ENABLE PERFORMANCE REPORTS)

        DEVM = 000017   NEW = <CR>              (KEEP DEFAULTS)

        24K MEMORY PRESENT
        PRINTER NOT PRESENT
        TERM #2 TESTING DROPPED

        INSERT SCRATCH DISKS, TYPE 'P' FOR NORMAL TESTING
        '240G' FOR NORMAL RESTARTS
        '250G' TO COPY SYS EXERCISER DISK
        '260G' FOR COMPATABLITIY  PASS 1: WRITE
        '270G' FOR                PASS 2: READ

(PROGRAM HALTS & WAITS FOR 'P' & CONTINUES....ABOVE NOTE & HALT OMITTED UNDER APT)

THE PROGRAM WILL BEGIN ACTUAL TESTING AT THIS POINT. (SEE SUMMARY OF TESTS SEC. 7.0)

DURING THE TESTS, THE PROGRAM WILL PRINT PROGRESS REPORTS (SEE SEC. 4.1) IF
BIT 12 IS SET IN THE SWREG (SEE SEC 2.3).
END OF PASS & TOTAL ERROR COUNT IS PRINTED AT THE CONCLUSION OF TESTING.
THE PROGRAM JUMPS TO THE BEGINNING & THE ABOVE SEQUENCE IS REPEATED UNTIL HALTED.
```

2.2 SPECIAL ENVIRONMENTS.

THIS DIAGNOSTIC FOLLOWS THE STANDARD PROCEDURE FOR RUNNING UNDER APT, RT-11 MONITORS, AS DESCRIBED IN THEIR RESPECTIVE PROCEDURES MANUAL AND SYSMAC PACKAGE.

2.3 OPERATIONAL SWITCH SETTINGS

THIS PROGRAM SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER (SWREG AT LOC. 176) FROM THE CONSOLE. THIS CAN BE ACCOMPLISHED BY DOING THE FOLLOWING:

1) TYPE CONTROL G <^G>: THIS WILL ALLOW THE CONSOLE TO ENTER DATA INTO LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

2) THE MACHINE WILL THEN TYPE:' SWR=XXXXXX NEW=' (XXXXXX IS THE OCTAL CONTENTS OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE 'NEW=' HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE OF THE FOLLOWING AT THE CONSOLE:

   A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A <CR>. (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED). LEADING ZEROS NEED NOT BE TYPED, AND IF MORE THAN 6 DIGITS ARE TYPED THE LAST 6 WILL BE USED. IF A <CR> IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH REGISTER CONTENTS WILL NOT BE CHANGED.

   B) IF A CONTROL U <^U> IS DEPRESSED THEN THE PROGRAM WILL SEND YOU BACK TO STEP 3.

   C) IF THE INPUT CHARACTER IS NOT ONE OF THE CHARACTERS MENTIONED ABOVE THEN A QUESTION MARK (?) WILL BE TYPED FOLLOWED BY A CARRAGE RETURN AND A LINE FEED SEQUENCE THEN PROCEED FROM STEP 3 (ERASING ALL PREVIOUS INPUT).

4) THE DIAGNOSTIC WILL CONTINUE ON TYPING <CR>.

SOFTWARE SWITCH REGISTER OPTIONS (SWREG)

BIT 15 SET = 100000 = HALT ON ERROR
    13 SET = 20000 = INHIBIT ERROR TYPEOUTS
    12 SET = 10000 = ENABLE PERFORMANCE REPORTS
    10 SET =   2000 = BELL ON ERROR

2.4 PROGRAM OPTIONS.

A DEVICE MAP ($DEVM) AT LOCATION 1246 (ENTERED BY A CONTROL-G & CONTROL-C)
IS EXAMINED BY THE PROGRAM TO DETERMINE THE METHOD OF TESTING
THE CLUSTER TERMINALS & THE COMM PORT.

THE DEFAULT VALUE = 000017 (INTERNAL LOOPBACK FOR COMM PORT & CLUSTER TERMINALS)
IT CAN BE CHANGED TO DO THE FOLLOWING:

$DEVM (LOC 1246) ENTERED BY CONTROL-G & C

```
BIT 15 SET=     100000= DROP PRINTER TESTS
    14 SET=      40000= DROP CLUSTER TERM #3 TESTS
    13 SET=      20000=                  #2
    12 SET=      10000=                  #1
    11 SET=       4000= DROP ASYNC COMM TESTS
    10 SET=       2000= DROP  SYNC COMM TESTS
     9 SET=       1000= DROP DRIVE 1 TESTS
     8 SET=        400= DROP DRIVE 0 TESTS
     7 SET=        200= DROP CLOCK TESTS

BIT  3 SET=         10= INT. LOOPBACK (MAINT MODE) FOR COMM PORT (DEFAULT)
     2 SET=          4= INT. LOOPBACK (MAINT MODE) FOR TERM #1   (DEFAULT)
     1 SET=          2= INT. LOOPBACK (MAINT MODE) FOR      #2   (DEFAULT)
     0 SET=          1= INT. LOOPBACK (MAINT MODE) FOR      #3   (DEFAULT)

BIT  3 CLR=             EXT. LOOPBACK FOR COMM PORT.
BIT  2 CLR=             EXT. LOOPBACK FOR TERM #1.
BIT  1 CLR=             EXT. LOOPBACK FOR TERM #2.
BIT  0 CLR=             EXT. LOOPBACK FOR TERM #3.
```

NOTES:

1. THE CONSOLE IS NOT TESTED.

2. BITS 15-7 SETTINGS WILL OVERRIDE BIT 3-0 SETTINGS.

3. THE PRINTER LOGIC WILL BE TESTED IF BIT 15 = 0 & THE DATA TERM READY
   IS ASSERTED ON THE CONNECTOR. (A PHYSICAL PRINTER IS NOT REQ'D)

4. A VT-50,52 MAY BE USED INSTEAD OF A PRINTER FOR A VISUAL CHECK.

2.5 EXECUTION TIMES.

ASSUMING ALL DEVICES & 2 DISK SUBSYSTEMS PRESENT:

1'ST PASS:

25 - 30 SEC TO STARTUP CLOCK, PRINTER, COMM DEVICE & CLUSTER TERMINALS.
10 - 40 SEC TO WRITE, READ & DATA COMPARE 1'ST TRACK ON BOTH DISKS.
5 - 60 SEC TO DO 10 RANDOM SEEKS ON EACH DISK BETWEEN 1'ST 10 TRACKS.

SUBSEQUENT PASSES:

30 SEC TO STARTUP CLOCK, PRINTER, COMM DEVICE & CLUSTER TERMINALS.
20 MIN TO WRITE, READ & DATA COMPARE ALL TRACKS ON BOTH DISKS.
5 MIN TO DO 500 RANDOM SEEKS ON EACH DISK BETWEEN ALL TRACKS.

2.6 POWER FAIL

THERE IS NO POWER FAIL AUTO-RESTART CAPABILITY IN THE PDT-11.

2.7 COMPATABILTIY TESTING

PASS 1: ENTERED FROM A '260G'.
WILL WRITE ALL TRACKS & SECTORS ON SELECTED DRIVES
& HALT AFTER AN OPERATOR PROMPT. TYPING 'P' WILL BEGIN PASS 2.

PASS 2: ENTERED BY A 'P' AFTER THE ABOVE HALT, OR A '270G'.
WILL PERFORM SEQUENTIAL & RANDOM READS ONLY, ON THE SELECTED DRIVES.

2.8 COPY UTILITY

TO PROVIDE BACKUP DISKS, A UTILITY IS PROVIDED IN THE PROGRAM TO COPY
THE SYSTEM EXERCISER DISK FROM DX0 TO DX1 IN THE FOLLOWING WAY:

1. BOOT THE EXERCISER NORMALLY FROM DX0.
2. ALLOW THE PROGRAM TO HALT AFTER THE WARNING MESSAGE TO USE SCRATCH DISKS.
3. AT THIS POINT, DO A '250G' TO ENTER THE COPY UTILITY.
4. THE OPERATOR WILL BE PROMPTED TO USE A SCRATCH DISK IN DX1 & TYPE 'P' TO PROCEED.
5. WHEN COMPLETED, THERE WILL BE A VERIFYING MESSAGE.
   THE OPERATOR CAN THEN DO A 'P' TO COPY ANOTHER IN THE SAME MANNER,
   OR A '240G' TO BEGIN NORMAL TESTING. (OPERATOR WILL BE PROMPTED)
6. THE COPY UTILITY CAN BE ENTERED AT ANY TIME BY DOING A 'BREAK' & '250G'.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

TYPICAL ERROR PRINTOUTS ARE SHOWN BELOW:

```
TERM #3 DATA COMP ERR
ERROR #    ERR PC       EXPECT  RECVD
000011     006516       000200  000100

DX1 SOFT ERR - DATA COMP
DXTST #    ERR PC       RXCS    RXES    RXSA    EXPECT  RECVD  # RETRIES
000003     010636       100337  000230  003405  100520  100120       4

DX0 HARD ERR - AFTER WRITE CMD
ERROR # DXTST #    ERR PC       RXCS    RXES    TRACK   SECTOR  #RETRIES
000016  000001     011246       100337  000230    59      18       10
```

WHERE ALL VALUES TYPED ARE OCTAL EXCEPT :
#RETRIES, TRACK, & SECTOR WHICH ARE IN DECIMAL.

BITS 15,13, & 10 OF THE SWITCH REGISTER (SWREG) CONTROL THE SEQUENCE OF EVENTS AFTER AN ERROR IS CAUGHT.

BIT 15 SET: CAUSES THE PROGRAM TO HALT IN THE ERROR ROUTINE. IF THE PROGRAM IS CONTINUED, IT WILL PROCEED FROM WHERE IT HALTED.

BIT 13 SET: DISABLES THE PRINTING OF THE ERROR MESSAGE.

BIT 10 SET: CAUSES THE BELL TO RING ON ERROR.

THE ERROR ROUTINE SUPPORTS THE CONTROL G <^G> FUNCTION. REFER TO SECTION 2.3 FOR DETAILS.

NOTE: SINCE THERE ARE NO SPECIFIC TEST NUMBERS, APT WILL REPORT ALL ERRORS AS TEST 0 ERRORS.

3.2 ERROR HALTS.

THE ONLY HALT IN THE EXERCISER IS IN THE ERROR ROUTINE, AND IS EXECUTED ONLY IF BIT 15 OF THE SWITCH REGISTER (SWREG) IS SET WHEN AN ERROR OCCURS.

4.0 PROGRESS & PERFORMANCE REPORTS

THE FOLLOWING REPORTS ARE ENABLED BIT 12 SET IN THE SWITCH REGISTER (SWREG LOC 176)
& WILL APPEAR FOLLOWING LOADING & STARTING AS DESCRIBED
IN SECTION 2.1 (ASSUMING ALL DEVICES ARE TO BE TESTED & NO ERRORS):

MEM TESTS DONE	(ALL MEMORY FROM LOCATION 0 TO THE BEGINNING OF THE MONITOR/ABS LOADER HAS BEEN TESTED.
CLOCK RUNNING	(100 INTERRUPTS HAVE BEEN DETECTED BEFORE EXERCISING THE NEXT DEVICE. THE CLOCK CONTINUES RUNNING IN INTERRUPT MODE.)
PRINTER RUNNING	(5 LINES HAVE BEEN PRINTED WITH ERROR BIT CHECKING ONLY. THE PRINTER RUNS CONTINUOUSLY IN INTERRUPT MODE AT 9600 BAUD. ITS ACTUAL PERFORMANCE IS A VISUAL CHECK IF PRINTER CONNECTED. NO OTHER CHECKING PERFORMED IF EXTERNAL LOOPBACK USED.
SYNC COMM DONE	(CHARACTERS 0 THRU 377 HAVE BEEN TRANSMITTED & RECEIVED WITH ODD PARITY ENABLED. INTERRUPTS ARE THEN DISABLED & THE SYNC COMM IS NO LONGER EXERCISED. THIS IS SO THAT THE ASYNC COMM CAN BE EXERCISED FOR THE DURATION OF THE PASS.)
ASYNC COMM RUNNING	(CHARS 0 THRU 377 HAVE BEEN TRANSMITTED & RECEIVED BEFORE EXERCISING THE NEXT DEVICE. IT CONTINUES RUNNING IN INTERRUPT MODE AT 9600 BAUD WITH ODD PARITY ENABLED & REPEATS THE 0 THRU 377 CYCLE)
TERM #1 RUNNING	(CHARACTERS 0 THRU 377 HAVE BEEN TRANSMITTED & RECEIVED BEFORE EXERXCISING THE NEXT DEVICE. THE TERMINAL KEEPS RUNNING CONTINUOUSLY AT 2400 BAUD IN INTERRUPT MODE & REPEATS THE 0 THRU 377 CYCLE)
TERM #2 RUNNING	(SAME AS #1)
TERM #3 RUNNING	(SAME AS #1)
DX0 TRK 0 DONE	(DRIVE 0, TRACK 0 IS WRITTEN WITH A DATA PATTERN, READ & DATA COMPARE PERFORMED)
DX1 TRK 0 DONE	(DRIVE 1, TRACK 0 IS WRITTEN WITH A DATA PATTERN, READ & DATA COMPARE PERFORMED)
DX0 TRK 1 DONE	(ETC)
DX1 TRK 1 DONE	(ETC UNTIL...)
DX0 DATA PATT DONE	(ALL TRACKS HAVE BEEN WRITTEN WITH A DATA PATTERN IN INTERRUPT MODE.)
DX1 DATA PATT DONE	(SAME AS DX0....TRK 0 ONLY FOR 1'ST PASS QUICK VERIFY)
DX0 RANDOM SEEKS DONE	(500 RANDOM SEEKS WITH READ & DATA COMPARE HAVE BEEN PERFORMED ON DRIVE 0)
DX1 RANDOM SEEKS DONE	(SAME AS DX0....10 RANDOM SEEKS ONLY FOR 1'ST PASS QUICK VERIFY)

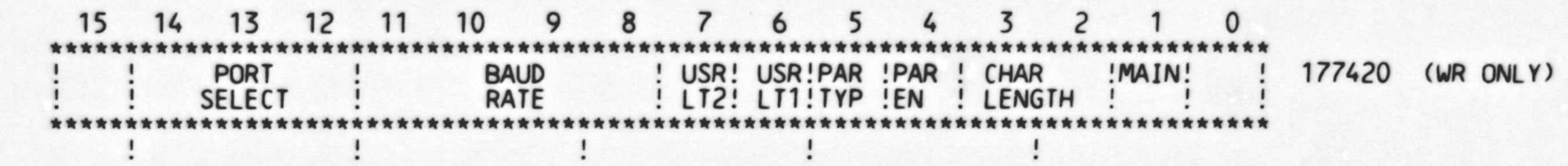
END OF PASS #1

TOTAL ERRORS: 0
TOTAL SOFT ERRORS: 0

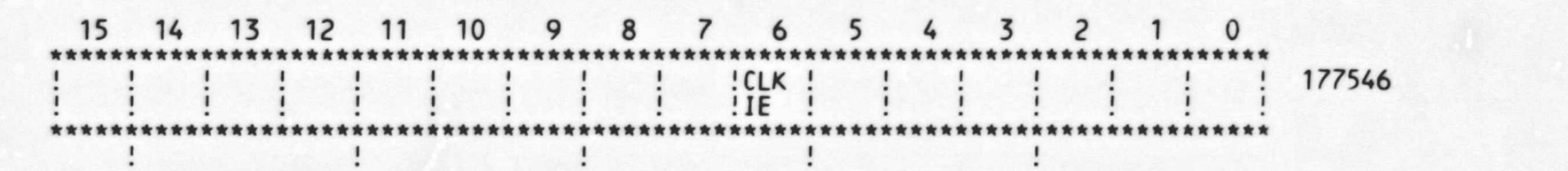
TOTAL ERRORS THIS PASS: 0
SOFT ERRORS THIS PASS: 0 (...& ENTIRE PROCESS REPEATS)

5.0 DEVICE REGISTERS.

PARAMETER REGISTER:



LINE CLOCK: VECTOR: 100



CONSOLE: ADDR: 177560-177566 VECTOR: 60/64
CLUSTER #1: 176500-176506 300/304
#2: 176510-176516 310/314
#3: 176520-176526 320/324

PRINTER: VECTOR: 200

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
PRS: ERR								PRIN RDY	PRIN IE							177514
PRB: PRINTER DATA BUFFER (WRITE ONLY)																177516

ASYNC MODEM: VECTOR: 330/334

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RCSR: DSI	RING	CTS	CAR DET		SEC REC	DSET RDY		RCVR DONE	RCVR IE	DSET IE		SEC XMIT	RTS	DTR		176610
RBUF: ERR	OR ERR	FR ERR	PAR ERR					RECEIVER DATA BUFFER								176612
XCSR:								XMIT RDY	XMIT IE						BRK	176614
XBUF: TRANSMITTER DATA BUFFER (WRITE ONLY)																176616

SYNC MODEM: VECTOR: 340/344

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RCSR: DSC	RING	CTS	CAR DET	REC ACT	SEC REC	DSET RDY	STRP SYNC	RCVR DONE	RCVR IE	DSET IE	SRCH SYNC	SEC XMIT	RTS	DTR		176620
RBUF: ERR	OR ERR		PAR ERR					RECEIVER DATA BUFFER								176622 (RD ONLY)
PCSR:	SYNC CHAR			WORD LENGTH	PAR IE	EVEN PAR	SYNC REG									176622 (WR ONLY)
XCSR: DNA			MA MODE	MA CLK			MAST RST	XMIT RDY	XMIT IE	DNA IE	SEND	H/F DUP				176624
XBUF: TRANSMITTER DATA BUFFER (WRITE ONLY)																176626

DISK: VECTOR: 264

	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
RXCS	ERR								CTL DONE	DRV IE		UNIT SEL	FUNCTION			GO	177170
RXDB	DATA BUFFER																177172
RXES									DRV RDY		DEL DATA	RNF	CRC	LOST DATA	INV ADDR	ID	177172
RXSA	TRACK 0 - 114(8)												SECTOR 1 - 32(8)				177174

6.0 DEVICE CONNECTOR LAYOUT

CLUSTER TERMINAL #3 PORT	COMM PORT
CLUSTER TERMINAL #2 PORT	PRINTER PORT
CLUSTER TERMINAL #1 PORT	CONSOLE PORT

B 2

7.0 SUMMARY OF TESTS (SEE PROGRAM LISTING FOR ADDT'L DETAILS ON SPECIFIC TESTS)

PHASE 1

MEMORY TESTS
A. LOC 0 THRU END OF PROGRAM IS TESTED FOR TIMEOUT.
B. END OF PROGRAM TO BOTTOM OF MONITOR/ABS LOADER IS TESTED WITH A 1010 & 0101 PATTERN.
C. IF 28K PRESENT, MEMORY FROM 28K TO 30K IS SIMILARLY TESTED.

PHASE 2

ALL AVAILABLE PERIPHERALS ARE EXERCISED CONTINUOUSLY IN INTERRUPT MODE.

SEE SECTION 2.4 FOR DETAILS OF SETTING UP THE DEVICE MAP ($DEVM) FOR DEVICE TO BE TESTED.
SEE SECTION 4.1 FOR DETAILS ON PROGRESS REPORTS.

START LINE CLOCK	INTERRUPTS ARE DETECTED.
START PRINTER	CONTINUOUS LINES ARE PRINTED.
START SYNC COMM PORT	CHARS 27 THRU 377 " " " 1 PASS ONLY
START ASYNC COMM PORT	CHARS 0 THRU 377 TESTED CONTINUOUSLY.
START CLUSTER TERMINAL #1	CHARS 0 THRU 377 ARE XMITTED, REC'D & TESTED.
START CLUSTER TERMINAL #2	SAME
START CLUSTER TERMINAL #3	SAME

PHASE 3

WHILE THE ABOVE DEVICES ARE INTERRUPTING RANDOMLY, DISK SUBSYSTEM TESTS BEGIN:

FILL & EMPTY BUFFER TESTS ARE PERFORMED TO VERIFY THAT NO CABLE CROSS-TALK PROBLEMS EXIST BEFORE BEGINNING ACTUAL DATA TRANSFERS TO THE DISK SURFACE.

DX0 & DX1 DATA PATTERN: WRITE, READ & DATA COMPARE 1'ST 10 TRACKS ON 1'ST PASS.
ALL TRACKS ON SUBSEQUENT PASSES.

DX0 & DX1 RANDOM SEEKS: READ & DATA COMPARE.
20 SEEKS ON 1'ST 10 TRACKS ON 1'ST PASS.
500 SEEKS ON ALL TRACKS ON SUBSEQUENT PASSES.

DX0 INITIALIZE, RESTORE, WRITE DELETED DATA, READ STATUS, & INVALID ADDRESS TESTS.

NOTE:

1. RECOVERY IS PERFORMED (RESTORE COMMAND) AFTER ANY RNF ERROR ON WRITE OR READ.
2. DATA IS TESTED AFTER A HARD CRC ERROR ON A READ TO TEST WHETHER DATA CRC ERROR OR CRC ERROR.

%

```
                                .TITLE  PDT-11 EXERCISER
                                ;*COPYRIGHT (C) 1978
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY GARY PAPAZIAN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*

                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;
                                ;       SWITCH          USE
                                ;       ------          ---
                                ;         15            HALT ON ERROR
                                ;         13            INHIBIT ERROR TYPEOUTS
                                ;         12            ENABLE PERFORMANCE REPORTS
                                ;         10            BELL ON ERROR
                                ;



                                .SBTTL  BASIC DEFINITIONS

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

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
000004                          R4=     %4              ;;GENERAL REGISTER
000005                          R5=     %5              ;;GENERAL REGISTER
000006                          R6=     %6              ;;GENERAL REGISTER
000007                          R7=     %7              ;;GENERAL REGISTER
000006                          SP=     %6              ;;STACK POINTER
000007                          PC=     %7              ;;PROGRAM COUNTER
```

```
                                ;*PRIORITY LEVEL DEFINITIONS
000000                          PR0=    0               ;;PRIORITY LEVEL 0
000040                          PR1=    40              ;;PRIORITY LEVEL 1
000100                          PR2=    100             ;;PRIORITY LEVEL 2
000140                          PR3=    140             ;;PRIORITY LEVEL 3
000200                          PR4=    200             ;;PRIORITY LEVEL 4
000240                          PR5=    240             ;;PRIORITY LEVEL 5
000300                          PR6=    300             ;;PRIORITY LEVEL 6
000340                          PR7=    340             ;;PRIORITY LEVEL 7

                                ;*''SWITCH REGISTER'' SWITCH DEFINITIONS
100000                          SW15=   100000
040000                          SW14=   40000
020000                          SW13=   20000
010000                          SW12=   10000
004000                          SW11=   4000
002000                          SW10=   2000
001000                          SW09=   1000
000400                          SW08=   400
000200                          SW07=   200
000100                          SW06=   100
000040                          SW05=   40
000020                          SW04=   20
000010                          SW03=   10
000004                          SW02=   4
000002                          SW01=   2
000001                          SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
100000                          BIT15=  100000
040000                          BIT14=  40000
020000                          BIT13=  20000
010000                          BIT12=  10000
004000                          BIT11=  4000
002000                          BIT10=  2000
001000                          BIT09=  1000
000400                          BIT08=  400
000200                          BIT07=  200
000100                          BIT06=  100
000040                          BIT05=  40
000020                          BIT04=  20
000010                          BIT03=  10
000004                          BIT02=  4
```

```
000002                  BIT01=  2
000001                  BIT00=  1
                        .EQUIV  BIT09,BIT9
                        .EQUIV  BIT08,BIT8
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC ''CPU'' TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;''T'' BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;''TRAP'' TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


                        ;;*********************************************************************
                        ;*      PDT-11/150 DEVICE VECTORS
                        ;;*********************************************************************

000200                  PRVEC=  200             ;PRINTER

000100                  CLKVEC= 100             ;LINE CLOCK

000330                  AMRVEC= 330             ;ASYNC MODEM RECVR
000334                  AMXVEC= 334             ;             XMITR

000340                  SMRVEC= 340             ;SYNC MODEM RECVR
000344                  SMXVEC= 344             ;            XMITR

000264                  RXVEC=  264             ;DISK

000300                  TK1VEC= 300             ;CLUSTER TERM #1 VECTORS
000304                  TP1VEC= 304
000310                  TK2VEC= 310             ;              #2
000314                  TP2VEC= 314
000320                  TK3VEC= 320             ;              #3
000324                  TP3VEC= 324
```

```
                        ;;*****************************************************************
                        ;       PDT-11/150 DEVICE REGISTERS
                        ;;*****************************************************************

177420          PARAM=  177420          ;PARAMETER REGISTER             (WRITE ONLY)

177546          CLK=    177546          ;LINE CLOCK

177514          PRS=    177514          ;PRINTER STATUS REG
177516          PRB=    177516          ;       BUFFER                  (WRITE ONLY)

176610          AMRC=   176610          ;ASYNC MODEM RECVR STATUS REG
176612          AMRB=   176612          ;               BUFFER
176614          AMXC=   176614          ;           XMITR STATUS REG
176616          AMXB=   176616          ;               BUFFER          (WRITE ONLY)

176620          SMRC=   176620          ;SYNC MODEM RECVR STATUS REG
176622          SMRB=   176622          ;               BUFFER          (READ ONLY)
176622          SMPAR=  176622          ;            PARAMETER REG      (WRITE ONLY)
176624          SMXC=   176624          ;           XMITR STATUS REG
176626          SMXB=   176626          ;               BUFFER          (WRITE ONLY)

177170          RXCS=   177170          ;DISK CSR
177172          RXDB=   177172          ;       DATA BUFF
177172          RXES=   177172          ;       ERROR & STATUS REG
177174          RXSA=   177174          ;       TRK & SECTOR ADDR REG

176500          TK1S=   176500          ;CLUSTER TERMINAL #1
176502          TK1B=   176502
176504          TP1S=   176504
176506          TP1B=   176506

176510          TK2S=   176510          ;               #2
176512          TK2B=   176512
176514          TP2S=   176514
176516          TP2B=   176516

176520          TK3S=   176520          ;               #3
176522          TK3B=   176522
176524          TP3S=   176524
176526          TP3B=   176526
```

```
                        ;;*****************************************************************
                        ;*      PARAMETER REGISTER EQUATES      WRITE ONLY
                        ;;*****************************************************************

                        ;BIT 14-12  PORT SELECT
                        ;
010000                  CONSOL= 10000           ;SELECT CONSOLE TERMINAL
000000                  CT1=    0               ;       CLUSTER TERMINAL #1
050000                  CT2=    50000           ;                       #2
060000                  CT3=    60000           ;                       #3
020000                  PRT=    20000           ;       PRINTER PORT
030000                  AMOD=   30000           ;       ASYNC COMM PORT
040000                  ULITE=  40000           ;       USER LIGHTS

                        ;BIT 11-8  BAUD RATE    (DELTA = 400)
                        ;
000000                  B50=    0               ;50 BAUD
000400                  B75=    400             ;75
001000                  B110=   1000            ;110
001400                  B134=   1400            ;134.5
002000                  B150=   2000            ;150
002400                  B300=   2400            ;300
003000                  B600=   3000            ;600
003400                  B1200=  3400            ;1200
004000                  B1800=  4000            ;1800
004400                  B2000=  4400            ;2000
005000                  B2400=  5000            ;2400 (CLUSTER DEFAULT)
005400                  B3600=  5400            ;3600
006000                  B4800=  6000            ;4800
006400                  B7200=  6400            ;7200
007000                  B9600=  7000            ;9600 (PRINTER & ASYNC DEFAULT)
007400                  B19200= 7400            ;19200
                        ;
                        ;IF 110 BAUD SELECTED, 2 STOP BITS ARE ASSUMED.

                        ;BITS 4-5 PARITY CONTROL
                        ;
000060                  EPAR=   60              ;EVEN PARITY ENABLE
000020                  OPAR=   20              ;ODD PARITY ENABLE (DEFAULT)

                        ;BITS 3-2  CHARACTER LENGTH (DELTA = 4)
                        ;
000000                  CHAR5=  0               ;5 BITS/CHAR
000004                  CHAR6=  4               ;6
000010                  CHAR7=  10              ;7
000014                  CHAR8=  14              ;8 (PRINTER,TERMINAL,MODEM DEFAULT)

                        ;BIT 1  MAINTENANCE BIT
                        ;
000002                  MAINT=  BIT1            ;ENABLE MAINT MODE
                                                ;PRINTER & DISK DO NOT HAVE MAINT MODE
```

```
                        ;;*****************************************************************
                        ;*      ASYNC MODEM BIT DEFINITIONS
                        ;;*****************************************************************

                        ;RCSR
                        ;
100000                  DSI=    BIT15           ;DATA SET INTERRUPT
040000                  RING=   BIT14           ;RING
020000                  CTS=    BIT13           ;CLEAR TO SEND
010000                  CDET=   BIT12           ;CARRIER DETECTED
002000                  SREC=   BIT10           ;SECONDARY RECD/SUPERVISORY RECD
001000                  DSRDY=  BIT9            ;DATA SET RDY
000200                  DONE=   BIT7            ;RECVR DONE
000100                  IE=     BIT6            ;RECVR IE
000040                  DSIE=   BIT5            ;DATA SET IE
000010                  SXMIT=  BIT3            ;SECONDARY XMIT/SUPERV XMIT
000004                  RTS=    BIT2            ;REQ TO SEND
000002                  DTR=    BIT1            ;DATA TERMINAL RDY

                        ;RBUF
                        ;
100000                  ERR=    BIT15           ;ERROR
040000                  ORERR=  BIT14           ;OVERRUN ERROR
020000                  FRERR=  BIT13           ;FRAMING ERROR
010000                  PARERR= BIT12           ;PARITY ERROR

                        ;XCSR
                        ;
000200                  RDY=    BIT7            ;XMIT RDY
000100                  IE=     BIT6            ;XMIT IE
000001                  BREAK=  BIT0            ;BREAK
```

```
                        ;;*****************************************************************
                        ;*      SYNC MODEM BIT DEFINITIONS
                        ;;*****************************************************************

                        ;RCSR
                        ;
100000                  DSC=    BIT15           ;DATA SET CHANGE
040000                  RING=   BIT14           ;RING
020000                  CTS=    BIT13           ;CLEAR TO SEND
010000                  CDET=   BIT12           ;CARRIER DETECTED
004000                  RACT=   BIT11           ;RECVR ACTIVE  (SYNC DETECT)
002000                  SREC=   BIT10           ;SECONDARY RECD/SUPERVISORY RECD
001000                  DSRDY=  BIT9            ;DATA SET RDY
000400                  STSYN=  BIT8            ;STRIP SYNC
000200                  DONE=   BIT7            ;RECVR DONE
000100                  IE=     BIT6            ;RECVR IE
000040                  DSIE=   BIT5            ;DATA SET IE
000020                  SRSYN=  BIT4            ;SEARCH SYNC
000010                  SXMIT=  BIT3            ;SECONDARY XMIT/SUPERV XMIT
000004                  RTS=    BIT2            ;REQ TO SEND
000002                  DTR=    BIT1            ;DATA TERMINAL RDY

                        ;RBUF
                        ;
100000                  ERR=    BIT15           ;ERROR
040000                  ORERR=  BIT14           ;OVERRUN ERROR
010000                  PARERR= BIT12           ;PARITY ERROR

                        ;PARCSR  PARAMETER CONT REG
                        ;
                        ;BIT 14   SYNC CHAR
                        ;
000000                  SYNC2=  0               ;2 SYNC CHARS (DEFAULT)
040000                  SYNC1=  BIT14           ;1

                        ;BITS 10-11     WORD LENGTH
                        ;
000000                  CHR5=   0               ;5 BITS/CHAR
002000                  CHR6=   2000            ;6
004000                  CHR7=   4000            ;7
006000                  CHR8=   6000            ;8 (DEFAULT)

                        ;BITS 8-9 PARITY CONTROL
                        ;
001400                  ENVPAR= 1400            ;ENABLE EVEN PARITY
001000                  ODDPAR= 1000            ;ENABLE ODD PARITY (DEFAULT)

                        ;XCSR
                        ;
100000                  DNA=    BIT15           ;DATA NOT AVAIL
010000                  MM=     BIT12           ;MAINT MODE
004000                  MCLK=   BIT11           ;MAINT CLOCK
000400                  MR=     BIT8            ;MASTER RESET
```

```
1007            000200                  RDY=    BIT7            ;XMIT RDY
1008            000100                  IE=     BIT6            ;XMIT IE
1009            000040                  DNAIE=  BIT5            ;DATA NOT AVAIL IE
1010            000020                  SEND=   BIT4            ;SEND
1011            000010                  HFDUP=  BIT3            ;1=HALF DUPLEX, 0=FULL DUPLEX (DEFAULT = 0 FOR TESTS)
1012
1013
1014
1015
1016                                    ;;*****************************************************************
1017                                    ;DISK BIT DEFINITIONS
1018                                    ;;*****************************************************************
1019
1020                                    ;RXCS
1021                                    ;
1022            100000                  ERR=    BIT15           ;ERROR
1023            000200                  DONE=   BIT7            ;CONTR RDY
1024            000100                  IE=     BIT6            ;DRIVE IE
1025            000020                  DX1=    BIT4            ;UNIT SELECT: 0=DX0, 1=DX1
1026            000020                  USEL=   BIT4            ;UNIT SELECT
1027
1028                                    ;FUNCTIONS (INCLUDES BIT0 = GO)
1029
1030            000001                  FBUF=   1               ;FILL BUFFER
1031            000003                  EBUF=   3               ;EMPTY BUFFER
1032            000005                  WSEC=   5               ;WRITE SECTOR
1033            000007                  RSEC=   7               ;READ SECTOR
1034            000011                  INITAL= 11              ;INITIALIZE
1035            000013                  RSTAT=  13              ;READ STATUS
1036            000015                  WDDSEC= 15              ;WRITE DELETED DATA SECTOR
1037            000017                  RESTOR= 17              ;RESTORE TO TRACK 0
1038
1039                                    ;RXES: ERROR & STATUS
1040                                    ;
1041            000200                  RDY=    BIT7            ;DRIVE RDY
1042            000040                  DD=     BIT5            ;DELETED DATA DET.
1043            000020                  RNF=    BIT4            ;RECORD NOT FOUND
1044            000010                  CRC=    BIT3            ;CRC ERROR
1045            000004                  LDATA=  BIT2            ;LOST DATA
1046            000002                  INVADR= BIT1            ;INVALID ADDR
1047            000001                  ID=     BIT0            ;INITIALIZE DONE
1048
1049
```

```
                                ;;*****************************************************************
                                ;*      SWITCH REGISTER
                                ;;*****************************************************************

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

        000100                          .=100
000100  000102                          102             ;SETUP CLOCK VECTOR AREA TO DO RTI
000102  000002                          2               ;IF ENABLED

        000200                          .=200
000200  000137  003174                  JMP     START   ;USE ONLY ONCE. WILL BE OVERLAID BY
                                                        ;PRINTER VECTOR ADDR.
        000240                          .=240
000240  000137  003174                  JMP     START   ;RESTART ADDR

        000250                          .=250
000250  000137  003156                  JMP     START3  ;ENTER HERE FOR COPY UTILITY
        000260                          .=260
000260  000137  003136                  JMP     START1  ;COMPATABILTIY PASS 1
        000270                          .=270
000270  000137  003144                  JMP     START2  ;               PASS 2


        001000          .=1000
                        .SBTTL  APT PARAMETER BLOCK

                        ;;*****************************************************************
                        ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
                        ;;*****************************************************************
        001000                  .$X=.   ;;SAVE CURRENT LOCATION
        000024                  .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
000024  000200                  200     ;;FOR APT START UP
        000044                  .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
000044  001000                  $APTHDR ;;POINT TO APT HEADER BLOCK
        001000                  .=.$X   ;;RESET LOCATION COUNTER
                        ;;*****************************************************************
                        ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
                        ;INTERFACE SPEC.

001000                  $APTHD:
001000  000000          $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
001002  001170          $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
001004  000024          $TSTM:  .WORD   20.     ;;RUN TIM OF LONGEST TEST
001006  000226          $PASTM: .WORD   150.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
001010  000000          $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
001012  000030                  .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

                        001100          .=1100
001100                  $CMTAG:                         ;;START OF COMMON TAGS
001100  000000                  .WORD   0
001102  000             $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001103  000             $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001104  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001106  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001110  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001112  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001114  000             $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001115  001             $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001116  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001120  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001122  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001124  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001126  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001130  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001132  000000                  .WORD   0
001134  000             $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001135  000             $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001136  000000                  .WORD   0
001140  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001142  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001144  177560          $TKS:   177560                  ;;TTY KBD STATUS
001146  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001150  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001152  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001154  000             $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001155  002             $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001156  012             $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A 'LINE FEED''
001157  000             $TPFLG: .BYTE   0               ;;'TERMINAL AVAILABLE'' FLAG (BIT<07>=0=YES)
001160  177607  000377  $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001164  077             $QUES:  .ASCII  /?/             ;;QUESTION MARK
001165  015             $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001166  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .EVEN
001170                  $MAIL:                          ;;APT MAILBOX
001170  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001172  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001174  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001176  000000          $PASS:  .WORD   APASS           ;;PASS COUNT
001200  000000          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001202  000000          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001204  000000          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
```

```
1155  001206  000000                $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
1156  001210                        $ETABLE:                 ;;APT ENVIRONMENT TABLE
1157  001210     000                $ENV:   .BYTE   AENV     ;;ENVIRONMENT BYTE
1158  001211     000                $ENVM:  .BYTE   AENVM    ;;ENVIRONMENT MODE BITS
1159  001212  000000                $SWREG: .WORD   ASWREG   ;;APT SWITCH REGISTER
1160  001214  000000                $USWR:  .WORD   AUSWR    ;;USER SWITCHES
1161  001216  000000                $CPUOP: .WORD   ACPUOP   ;;CPU TYPE,OPTIONS
1162                                ;*                       BITS 15-11=CPU TYPE
1163                                ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1164                                ;*                              11/70=06,PDQ=07,Q=10
1165                                ;*                       BIT 10=REAL TIME CLOCK
1166                                ;*                       BIT  9=FLOATING POINT PROCESSOR
1167                                ;*                       BIT  8=MEMORY MANAGEMENT
1168  001220     000                $MAMS1: .BYTE   AMAMS1   ;;HIGH ADDRESS,M.S. BYTE
1169  001221     000                $MTYP1: .BYTE   AMTYP1   ;;MEM. TYPE,BLK#1
1170                                ;*                       MEM.TYPE BYTE   --  (HIGH BYTE)
1171                                ;*                              900 NSEC CORE=001
1172                                ;*                              300 NSEC BIPOLAR=002
1173                                ;*                              500 NSEC MOS=003
1174  001222  000000                $MADR1: .WORD   AMADR1   ;;HIGH ADDRESS,BLK#1
1175                                ;*                         MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
1176  001224     000                $MAMS2: .BYTE   AMAMS2   ;;HIGH ADDRESS,M.S. BYTE
1177  001225     000                $MTYP2: .BYTE   AMTYP2   ;;MEM.TYPE,BLK#2
1178  001226  000000                $MADR2: .WORD   AMADR2   ;;MEM.LAST ADDRESS,BLK#2
1179  001230     000                $MAMS3: .BYTE   AMAMS3   ;;HIGH ADDRESS,M.S.BYTE
1180  001231     000                $MTYP3: .BYTE   AMTYP3   ;;MEM.TYPE,BLK#3
1181  001232  000000                $MADR3: .WORD   AMADR3   ;;MEM.LAST ADDRESS,BLK#3
1182  001234     000                $MAMS4: .BYTE   AMAMS4   ;;HIGH ADDRESS,M.S.BYTE
1183  001235     000                $MTYP4: .BYTE   AMTYP4   ;;MEM.TYPE,BLK#4
1184  001236  000000                $MADR4: .WORD   AMADR4   ;;MEM.LAST ADDRESS,BLK#4
1185  001240  000300                $VECT1: .WORD   AVECT1   ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1186  001242  000000                $VECT2: .WORD   AVECT2   ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1187  001244  176500                $BASE:  .WORD   ABASE    ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1188  001246  000017                $DEVM:  .WORD   ADEVM    ;;DEVICE MAP
1189  001250                        $ETEND:
1190                                .MEXIT
```

```
                                                .SBTTL  ERROR POINTER TABLE

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

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


001250                                          $ERRTB:
                                                ;;              GLOBAL DATA
                                                ;ERR 1
001250  020322  023160  023566                          EM1,DH2,DT2,DF
001256  024062
                                                ;ERR 2
001260  020322  023160  023566                          EM1,DH2,DT2,DF
001266  024062
                                                ;ERR 3
001270  020341  023204  023576                          EM2,DH3,DT3,DF
001276  024062
                                                ;ERR 4
001300  020322  023160  023566                          EM1,DH2,DT2,DF
001306  024062
                                                ;ERR 5
001310  020341  023204  023576                          EM2,DH3,DT3,DF
001316  024062
                                                ;ERR 6
001320  020363  023141  023560                          EM3,DH1,DT1,DF
001326  024062
                                                ;ERR 7
001330  020427  023246  023612                          EM4,DH4,DT4,DF
001336  024062
                                                ;ERR 10
001340  020452  023277  023622                          EM5,DH5,DT5,DF
001346  024062
                                                ;ERR 11
001350  020500  023277  023634                          EM6,DH5,DT6,DF
001356  024062
                                                ;ERR 12
001360  020526  023277  023646                          EM7,DH5,DT7,DF
001366  024062
                                                ;ERR 13
001370  020554  023277  023660                          EM8,DH5,DT8,DF
001376  024062
                                                ;ERR 14
001400  020605  023277  023660                          EM9,DH5,DT8,DF
001406  024062
                                                ;ERR 15
001410  020635  023334  023672                          EM10,DH6,DT9,DF
```

```
 1245  001416  024062
 1246                                  ;ERR 16
 1247  001420  020662  023401  023710          EM11,DH7,DT10,DF1
 1248  001426  024070
 1249                                  ;ERR 17
 1250  001430  020662  023401  023710          EM11,DH7,DT10,DF1
 1251  001436  024070
 1252                                  ;ERR 20
 1253  001440  020700  023401  023710          EM13,DH7,DT10,DF1
 1254  001446  024070
 1255                                  ;ERR 21
 1256  001450  020731  023401  023710          EM14,DH7,DT10,DF1
 1257  001456  024070
 1258                                  ;ERR 22
 1259  001460  020762  023470  023732          EM15,DH8,DT11,DF2
 1260  001466  024100
 1261                                  ;ERR 23
 1262  001470  021013  023470  023732          EM16,DH8,DT11,DF2
 1263  001476  024100
 1264                                  ;ERR 24
 1265  001500  021044  023401  023754          EM17,DH7,DT12,DF1
 1266  001506  024070
 1267                                  ;ERR 25
 1268  001510  021076  023401  023754          EM18,DH7,DT12,DF1
 1269  001516  024070
 1270                                  ;ERR 26
 1271  001520  021130  023401  023754          EM19,DH7,DT12,DF1
 1272  001526  024070
 1273                                  ;ERR 27
 1274  001530  021161  023401  023754          EM20,DH7,DT12,DF1
 1275  001536  024070
 1276                                  ;ERR 30
 1277  001540  021212  023470  023776          EM21,DH8,DT13,DF2
 1278  001546  024100
 1279                                  ;ERR 31
 1280  001550  021243  023470  023776          EM22,DH8,DT13,DF2
 1281  001556  024100
 1282                                  ;ERR 32
 1283  001560  021274  023141  023560          EM23,DH1,DT1,DF
 1284  001566  024062
 1285                                  ;ERR 33
 1286  001570  021305  023246  023612          EM24,DH4,DT4,DF
 1287  001576  024062
 1288                                  ;ERR 34
 1289  001600  021322  023141  023560          EM25,DH1,DT1,DF
 1290  001606  024062
 1291                                  ;ERR 35
 1292  001610  021337  023141  023560          EM26,DH1,DT1,DF
 1293  001616  024062
 1294                                  ;ERR 36
 1295  001620  021354  023141  023560          EM27,DH1,DT1,DF
 1296  001626  024062
```

```
1297                                        ;ERR 37
1298  001630  021371  023141  023560              EM28,DH1,DT1,DF
1299  001636  024062
1300                                        ;ERR 40
1301  001640  021410  023141  023560              EM29,DH1,DT1,DF
1302  001646  024062
1303                                        ;ERR 41
1304  001650  021430  023334  024020              EM30,DH6,DT14,DF
1305  001656  024062
1306                                        ;ERR 42
1307  001660  021441  023334  024020              EM31,DH6,DT14,DF
1308  001666  024062
1309                                        ;ERR 43
1310  001670  021452  023141  023560              EM32,DH1,DT1,DF
1311  001676  024062
1312                                        ;ERR 44
1313  001700  021515  023141  023560              EM33,DH1,DT1,DF
1314  001706  024062
1315                                        ;ERR 45
1316  001710  021556  023141  023560              EM34,DH1,DT1,DF
1317  001716  024062
1318                                        ;ERR 46
1319  001720  021430  023334  024020              EM30,DH6,DT14,DF
1320  001726  024062
1321                                        ;ERR 47
1322  001730  021430  023334  024020              EM30,DH6,DT14,DF
1323  001736  024062
1324                                        ;ERR 50
1325  001740  021430  023334  024020              EM30,DH6,DT14,DF
1326  001746  024062
1327                                        ;ERR 51
1328  001750  021430  023334  024020              EM30,DH6,DT14,DF
1329  001756  024062
1330                                        ;ERR 52
1331  001760  021615  023334  023672              EM35,DH6,DT9,DF
1332  001766  024062
1333                                        ;ERR 53
1334  001770  021662  023470  023732              EM36,DH8,DT11,DF2
1335  001776  024100
1336                                        ;ERR 54
1337  002000  021731  023334  023672              EM37,DH6,DT9,DF
1338  002006  024062
1339                                        ;ERR 55
1340  002010  021755  023334  023672              EM38,DH6,DT9,DF
1341  002016  024062
1342                                        ;ERR 56
1343  002020  021430  023334  024020              EM30,DH6,DT14,DF
1344  002026  024062
1345                                        ;ERR 57
1346  002030  022023  023334  023672              EM39,DH6,DT9,DF
1347  002036  024062
1348                                        ;ERR 60
1349  002040  021430  023334  024020              EM30,DH6,DT14,DF
1350  002046  024062
```

```
1351                                        ;ERR 61
1352  002050  022053  023334  023672              EM40,DH6,DT9,DF
1353  002056  024062
1354                                        ;ERR 62
1355  002060  022116  023334  023672              EM41,DH6,DT9,DF
1356  002066  024062
1357                                        ;ERR 63
1358  002070  022152  023334  023672              EM42,DH6,DT9,DF
1359  002076  024062
1360                                        ;ERR 64
1361  002100  022173  023401  023710              EM43,DH7,DT10,DF1
1362  002106  024070
1363                                        ;ERR 65
1364  002110  022173  023401  023710              EM43,DH7,DT10,DF1
1365  002116  024070
1366                                        ;ERR 66
1367  002120  022225  023401  023710              EM45,DH7,DT10,DF1
1368  002126  024070
1369                                        ;ERR 67
1370  002130  022263  023401  023710              EM46,DH7,DT10,DF1
1371  002136  024070
1372                                        ;ERR 70
1373  002140  022321  023401  023754              EM47,DH7,DT12,DF1
1374  002146  024070
1375                                        ;ERR 71
1376  002150  022360  023401  023754              EM48,DH7,DT12,DF1
1377  002156  024070
1378                                        ;ERR 72
1379  002160  022417  023401  023754              EM49,DH7,DT12,DF1
1380  002166  024070
1381                                        ;ERR 73
1382  002170  022455  023401  023754              EM50,DH7,DT12,DF1
1383  002176  024070
1384                                        ;ERR 74
1385  002200  022513  023334  023672              EM51,DH6,DT9,DF
1386  002206  024062
1387                                        ;ERR 75
1388  002210  022537  023334  024020              EM52,DH6,DT14,DF
1389  002216  024062
1390                                        ;ERR 76
1391  002220  022563  023277  024036              EM53,DH5,DT15,DF
1392  002226  024062
1393                                        ;ERR 77
1394  002230  022563  023277  024050              EM53,DH5,DT16,DF
1395  002236  024062
```

```
1396
1397                                    ; ERRORS FOR COPY UTILITY
1398
1399                                    ;ERR 100
1400  002240  020700  023334  024020              EM13,DH6,DT14,DF
1401  002246  024062
1402                                    ;ERR 101
1403  002250  021044  023334  024020              EM17,DH6,DT14,DF
1404  002256  024062
1405                                    ;ERR 102
1406  002260  022632  023334  024020              EM54,DH6,DT14,DF
1407  002266  024062
1408                                    ;ERR 103
1409  002270  021130  023334  024020              EM19,DH6,DT14,DF
1410  002276  024062
1411
1412                                    ;BAD SECTOR/TRACK TABLE FULL ERRORS
1413
1414                                    ;ERR 104
1415  002300  022647  000000  000000              EM55,0,0,0
1416  002306  000000
1417                                    ;ERR 105
1418  002310  022704  000000  000000              EM56,0,0,0
1419  002316  000000
1420
1421                                    ;CRC ERRORS
1422
1423                                    ;ERR 106
1424  002320  022741  023401  023710              EM57,DH7,DT10,DF1
1425  002326  024070
1426                                    ;ERR 107
1427  002330  022762  023470  023732              EM58,DH8,DT11,DF2
1428  002336  024100
1429                                    ;ERR 110
1430  002340  023003  000000  000000              EM59,0,0,0
1431  002346  000000
1432                                    ;ERR 111
1433  002350  023041  023401  023754              EM60,DH7,DT12,DF1
1434  002356  024070
1435                                    ;ERR 112
1436  002360  023062  023470  023776              EM61,DH8,DT13,DF2
1437  002366  024100
1438                                    ;ERR 113
1439  002370  023103  000000  000000              EM62,0,0,0
1440  002376  000000
```

```
                                ;;*****************************************************************
                                ;REGISTERS & BUFFERS FCR DX0 TESTS
                                ;;*****************************************************************

002400  000000          D0PAT:  0               ;DX0 PATT TEST DONE FLAG
002402  000000          D0CNT:  0               ;DX0 RANDOM SEEK COUNTER
002404  000000          D0ERR:  0               ;DX0 RCSR ERR COUNTER
002406  000000          D0TRK:  0               ;DX0 TRACK
002410  000000          D0SEC:  0               ;DX0 SECTOR
002412  000000          D0CMER: 0               ;DX0 DATA COMPARE ERROR FLAG
002414  000000          D0CERR: 0               ;DX0 TOTAL DATA COMP ERROR CT
002416  000000          D0EXP:  0               ;DX0 EXPECTED DATA
002420  000000          D0REC:  0               ;DX0 RECEIVED DATA
002422  000000          D0CRC:  0               ;DX0 CRC ERROR FLAG
002424  000100          D0BUF:  .BLKW 100       ;DX0 DATA BUFFER
002624  000012          D0BAD:  .BLKW 10.       ;DX0 BAD TRK/SEC TABLE

                                ;;*****************************************************************
                                ;REGISTERS & BUFFERS FOR DX1 TESTS
                                ;;*****************************************************************

002650  000000          D1PAT:  0               ;DX1 PATT TEST DONE FLAG
002652  000000          D1CNT:  0               ;DX1 RANDOM SEEK COUNTER
002654  000000          D1ERR:  0               ;DX1 RCSR ERR COUNTER
002656  000000          D1TRK:  0               ;DX1 TRACK
002660  000000          D1SEC:  0               ;DX1 SECTOR
002662  000000          D1CMER: 0               ;DX1 DATA COMPARE ERROR FLAG
002664  000000          D1CERR: 0               ;DX1 TOTAL DATA COMPARE ERROR CT
002666  000000          D1EXP:  0               ;DX1 EXPECTED DATA
002670  000000          D1REC:  0               ;DX1 RECEIVED DATA
002672  000000          D1CRC:  0               ;DX1 CRC ERROR FLAG
002674  000100          D1BUF:  .BLKW 100       ;DX1 DATA BUFFER
003074  000012          D1BAD:  .BLKW 10.       ;DX1 BAD TRK/SEC TABLE

003120  000000          DXTST:  0               ;1 = DATA PATT TEST
                                                ;2 = RANDOM SEEK TEST
                                                ;3 = INITIALIZE TEST
                                                ;4 = RESTORE TEST
                                                ;5 = WRITE DELETED DATA TEST
                                                ;6 = INVALID ADDRESS TEST

003122  000000          FIN:    0               ;COMMON FLG FOR DX0 & DX1 FOR TRK/SEC FINISHED
003124  000000          MAXTRK: 0               ;MAX TRACK #
                                                ;0 FOR 1'ST PASS, 114 FOR SUBSEQUENT PASSES. SET AT START.
003126  000000          MAXSK:  0               ;# OF RANDOM SEEKS TO BE PERFORMED.
                                                ;10 FOR 1'ST PASS, 500 FOR SUBSEQUENT PASSES. SET AT EOP.
003130  000000          COPFLG: 0               ; 0 = NORM TEST   1 = COPY UTILITY ACTIVE
003132  000000          COMP1:  0               ;1 = COMPAT TESTS, PASS 1
003134  000000          COMP2:  0               ;1 =               PASS 2
```

```
1493                                        .SBTTL  PROGRAM
1494
1495  003136  005237  003132          START1: INC     COMP1           ;COMPAT PASS 1
1496  003142  000416                          BR      S1
1497
1498  003144  005237  003134          START2: INC     COMP2           ;COMPAT PASS 2
1499  003150  005037  003132                  CLR     COMP1
1500  003154  000413                          BR      S2
1501
1502  003156  005237  003130          START3: INC     COPFLG          ;COPY UTILITY
1503  003162  005037  003132                  CLR     COMP1
1504  003166  005037  003134                  CLR     COMP2
1505  003172  000406                          BR      S3
1506
1507  003174  005037  003132          START:  CLR     COMP1           ;NORMAL TESTING
1508  003200  005037  003134          S1:     CLR     COMP2
1509  003204  005037  003130          S2:     CLR     COPFLG
1510
1511  003210  012706  001100          S3:     MOV     #STACK,SP       ;SETUP STACK POINTER
1512  003214  042777  000100  175722          BIC     #100,@$TKS      ;DISABLE ANY TTY INTERRUPTS
1513                                                                  ;TO RT-11 MONITOR
1514                                          ;MTPS   #PR4            ;DISABLE INTERRUPTS
1515  003222  106427                          .WORD   106427
1516  003224  000200                          PR4
1517  003226  013737  000042  004440          MOV     42,HLD42        ;SAVE
1518  003234  123727  001210  000001          CMPB    $ENV,#1         ;APT?
1519  003242  001402                          BEQ     2$              ;BR IF YES
1520  003244  005037  000042                  CLR     42              ;ELSE CLR SO WE CAN USE SOFTSWR
1521  003250                          2$:
1522                                        .SBTTL  INITIALIZE THE COMMON TAGS
1523                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
1524  003250  012706  001100                  MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
1525  003254  005026                          CLR     (R6)+           ;;CLEAR MEMORY LOCATION
1526  003256  022706  001140                  CMP     #SWR,R6 ;;DONE?
1527  003262  001374                          BNE     .-6             ;;LOOP BACK IF NO
1528  003264  012706  001100                  MOV     #STACK,SP       ;;SETUP THE STACK POINTER
1529                                        ;;INITIALIZE A FEW VECTORS
1530  003270  012737  025502  000030          MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
1531  003276  012737  000340  000032          MOV     #340,@#EMTVEC+2 ;;LEVEL 7
1532  003304  012737  026534  000034          MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
1533  003312  012737  000340  000036          MOV     #340,@#TRAPVEC+2;LEVEL 7
1534                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
1535                                        ;;EQUAL TO A ''-1'', SETUP FOR A SOFTWARE SWITCH REGISTER.
1536  003320  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
1537  003324  012737  003360  000004          MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
1538  003332  012737  177570  001140          MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
1539  003340  012737  177570  001142          MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
1540  003346  022777  177777  175564          CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
1541  003354  001012                          BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
1542                                                                  ;;AND  THE HARDWARE SWR IS NOT = -1
1543  003356  000403                          BR      65$             ;;BRANCH IF NO TIMEOUT
1544  003360  012716  003366          64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
1545  003364  000002                          RTI
1546  003366  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
```

```
1547  003374  012737  000174  001142          MOV     #DISPREG,DISPLAY
1548  003402  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1549
1550  003406  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
1551  003412  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1552  003420  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1553  003422  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1554  003430                          67$:
1555                                  .SBTTL  TYPE PROGRAM NAME
1556                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1557  003430  005227  177777                  INC     #-1             ;;FIRST TIME?
1558  003434  001042                          BNE     68$             ;;BRANCH IF NO
1559  003436  104401  003504                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1560                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1561  003442  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1562  003446  001012                          BNE     70$             ;;BRANCH IF YES
1563  003450  123727  001210  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1564  003456  001406                          BEQ     70$             ;;BRANCH IF YES
1565  003460  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1566  003466  001005                          BNE     71$             ;;BRANCH IF NO
1567  003470  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1568  003472  000403                          BR      71$
1569  003474  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1570  003502                          71$:
1571  003502  000417                          BR      68$             ;;GET OVER THE ASCIZ
1572                                  ;;69$:  .ASCIZ  <CRLF>*PDT-11/150 SYSTEM EXERCISER*<CRLF>
1573  003542                          68$:
1574  003542  005737  003130                  TST     COPFLG          ;COPY UTILITY?
1575  003546  001057                          BNE     LOOP            ;BR IF YES
1576
1577  003550  005737  003132                  TST     COMP1           ;COMPAT PASS 1?
1578  003554  001054                          BNE     LOOP            ;BR IF YES
1579  003556  005737  003134                  TST     COMP2           ;COMPAT PASS 2?
1580  003562  001051                          BNE     LOOP            ;BR IF YES
1581
1582  003564  012737  000000  003124          MOV     #0,MAXTRK       ;TRK 0 ONLY FOR 1'ST PASS (PATCH-ABLE)
1583  003572  012737  000012  003126          MOV     #10.,MAXSK      ;10 RANDOM SEEKS FOR 1'ST PASS
1584
1585  003600  005037  026052                  CLR     TERR            ;TOTAL ERR CT
1586  003604  005037  026054                  CLR     TSERR           ;TOTAL SOFT ERR CT FOR EOP TYPEOUT
1587
1588  003610  052737  000006  176610          BIS     #<RTS!DTR>,AMRC
1589  003616  042737  000006  176610          BIC     #<RTS!DTR>,AMRC ;DTR MUST BE TOGGLED.
```

```
1590
1591  003624  005227  177777            INC     #-1             ;1'ST TIME?
1592  003630  001005                    BNE     4$              ;BR IF NO
1593  003632  123727  001210  000001    CMPB    $ENV,#1         ;APT?
1594  003640  001401                    BEQ     4$              ;BR IF YES
1595  003642  104412                    GTDEVM                  ;SHOW CURRENT DEVM & GET NEW
1596
1597  003644  004737  010256      4$:   JSR     PC,SIZE         ;SEE WHO IS ON THE SYSTEM
1598  003650  123727  001210  000001    CMPB    $ENV,#1         ;ARE WE RUNNING UNDER APT?
1599  003656  001413                    BEQ     LOOP            ;BR IF YES & DONT HALT
1600  003660  032737  001000  001246    BIT     #1000,$DEVM     ;DX0 THERE?
1601  003666  001404                    BEQ     1$              ;BR IF YES
1602  003670  032737  000400  001246    BIT     #400,$DEVM      ;DX1 THERE?
1603  003676  001003                    BNE     LOOP            ;BR IF NO
1604  003700  104401  017215      1$:   TYPE    ,WARNING        ;INSERT SCRATCH DISKS
1605  003704  000000                    HALT
1606
1607
1608  003706  012706  001100      LOOP: MOV     #1100,SP        ;LOOP HERE AFTER EOP & RESET STACK
1609
1610  003712  004737  011430            JSR     PC,TIMER1       ;TIMER TO ALLOW 'P' TO PRINT BEFORE RESET
1611
1612  003716  000005                    RESET                   ;CLEAR ALL INTERRUPT ENABLES
1613  003720  042737  000100  177546    BIC     #IE,CLK         ;RESET SETS CLK IE. DONT ALLOW YET
1614
1615                                    ;MTPS   #PR0            ;ALLOW INTERRUPTS
1616  003726  106427                    .WORD   106427
1617  003730  000000                    PR0
1618
1619  003732  052737  000006  176610    BIS     #<RTS!DTR>,AMRC ;TO CONDITION ALL TERMINALS &
1620                                                            ;COMM PORT TO OPERATE IN EXT LOOPBACK MODE.
1621                                                            ;NO HARM IF LOOPBACK NOT USED.
1622  003740  004737  011430            JSR     PC,TIMER1       ;TIMER TO ALLOW PREV XFERS TO FINISH
1623
1624  003744  005037  026056            CLR     PERR            ;PASS ERR COUNT FOR EOP TYPEOUT
1625  003750  005037  026060            CLR     PSERR           ;PASS SOFT ERR COUNT FOR EOP TYPEOUT
1626
1627  003754  004737  011526            JSR     PC,CLRBAD       ;CLEAR BAD SECTOR/TRACK TABLES
1628
1629  003760  005737  003132            TST     COMP1           ;COMPAT PASS 1?
1630  003764  001004                    BNE     1$              ;BR IF YES
1631  003766  005737  003134            TST     COMP2           ;COMPAT PASS 2?
1632  003772  001001                    BNE     1$              ;BR IF YES
1633  003774  000405                    BR      2$              ;ELSE TEST FOR COPY
1634
1635  003776  012737  000114  003124  1$:   MOV     #114,MAXTRK
1636  004004  000137  006300            JMP     DXTST1          ;COMPATABILITY TESTING
1637
1638  004010  005737  003130      2$:   TST     COPFLG          ;COPY UTILITY?
1639  004014  001402                    BEQ     MEMTST          ;BR IF NO...NORMAL TESTING
1640  004016  000137  016220            JMP     COPY            ;ELSE GO COPY
1641
```

```
                                        .SBTTL  PROGRAM
                                        ;;*****************************************************************
                                        ;       MEMORY TESTS
                                        ;
                                        ;MEMORY FROM 0 TO THE LAST LOC OF PROGRAM (LSTAD) IS TESTED FOR TIMEOUT.
                                        ;MEMORY FROM 'LSTAD' TO THE BOTTOM OF THE RT11 MONITOR/ABS LOADER
                                        ;IS TESTED BY A PASS OF A 1010 PATT FOLLOWED BY A PASS OF 0101 PATTERN.
                                        ;IF THE SYSTEM HAS A 28K MEMORY, 28K TO 30K WILL BE TESTED SIMILARLY.
                                        ;;*****************************************************************

004022  004737  010744          MEMTST: JSR     PC,EXAMKB

004026  012737  012004  000004          MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
004034  012737  000200  000006          MOV     #200,ERRVEC+2

004042  013703  010630                  MOV     MAXMEM,R3
004046  162703  000300                  SUB     #300,R3         ;TEST ONLY AS FAR AS ABS LOADER IF NOT RT-11

004052  005004                          CLR     R4              ;ADDR POINTER
004054  005037  012012          1$:     CLR     INTFLG
004060  005714                          TST     (R4)            ;TEST LOCS 0 THRU END OF PGM.
004062  005737  012012                  TST     INTFLG          ;TIMEOUT?
004066  001403                          BEQ     2$              ;BR IF NO
004070  010437  004436                  MOV     R4,ADDR         ;FOR ERR TYP
004074  104001                          ERROR   1               ;TIMEOUT ON READ
004076  062704  000002          2$:     ADD     #2,R4           ;BUMP ADDR
004102  020427  026616                  CMP     R4,#LSTAD       ;AT END?
004106  001362                          BNE     1$              ;BR IF NO

004110  005037  004430                  CLR     MEMCT
004114  012737  125252  004434          MOV     #125252,PAT     ;DATA PATT FOR 1'ST PASS
004122  012704  026616          8$:     MOV     #LSTAD,R4       ;ADDR POINTER. R/W LOCS LSTAD THRU
                                                                ;BOT OF RT11 OR BOT OF ABS LOADER
004126  005037  012012          3$:     CLR     INTFLG
004132  013714  004434                  MOV     PAT,(R4)        ;WRITE
004136  005737  012012                  TST     INTFLG          ;TIMEOUT?
004142  001403                          BEQ     4$              ;BR IF NO
004144  010437  004436                  MOV     R4,ADDR         ;FOR ERR TYP
004150  104002                          ERROR   2               ;TIMEOUT ON WRITE
004152  011437  004432          4$:     MOV     (R4),MEMHLD     ;READ
004156  023737  004432  004434          CMP     MEMHLD,PAT      ;COMPARE
004164  001403                          BEQ     6$
004166  010437  004436                  MOV     R4,ADDR         ;FOR ERR TYP
004172  104003                          ERROR   3               ;COMPARE ERROR

004174  062704  000002          6$:     ADD     #2,R4           ;BUMP POINTER
004200  023727  004440  001000          CMP     HLD42,#1000     ;RT11 IF LOC 42 = 1000
004206  001403                          BEQ     7$              ;BR IF RT11

004210  020403                          CMP     R4,R3           ;ELSE AT BOT OF ABS LOADER?
004212  001345                          BNE     3$              ;BR IN NO
004214  000403                          BR      5$              ;ELSE EXIT
```

```
1694
1695  004216  020437  000054          7$:   CMP   R4,54            ;LOC 54 CONTAINS LOW ADDR OF RT11 MONITOR
1696  004222  001341                        BNE   3$                ;BR IN NOT THERE
1697  004224  005737  004430          5$:   TST   MEMCT
1698  004230  001006                        BNE   9$
1699  004232  005237  004430                INC   MEMCT
1700  004236  012737  052525  004434        MOV   #052525,PAT       ;DATA PATT FOR 2'ND PASS
1701  004244  000726                        BR    8$                ;REPEAT
1702
1703  004246  023727  010630  157776  9$:   CMP   MAXMEM,#157776    ;DO WE HAVE 28K?
1704  004254  001047                        BNE   15$               ;BR IF NO & EXIT
1705
1706  004256  012737  125252  004434        MOV   #125252,PAT       ;DATA PATT FOR 1'ST PASS OF HI MEM
1707  004264  012704  160000          10$:  MOV   #160000,R4        ;ADDR PTR TO TOP OF 28K
1708  004270  005037  012012          11$:  CLR   INTFLG
1709  004274  013714  004434                MOV   PAT,(R4)          ;WRITE
1710  004300  005737  012012                TST   INTFLG
1711  004304  001403                        BEQ   12$
1712  004306  010437  004436                MOV   R4,ADDR           ;FOR ERR TYP
1713  004312  104004                        ERROR 4                 ;TIMEOUT
1714  004314  011437  004432          12$:  MOV   (R4),MEMHLD       ;READ
1715  004320  023737  004432  004434        CMP   MEMHLD,PAT        ;COMPARE
1716  004326  001403                        BEQ   14$
1717  004330  010437  004436                MOV   R4,ADDR           ;FOR ERR TYP
1718  004334  104005                        ERROR 5                 ;COMPARE ERROR
1719  004336  062704  000002          14$:  ADD   #2,R4
1720  004342  020427  167776                CMP   R4,#167776        ;AT END OF 30K?
1721  004346  001350                        BNE   11$               ;BR IF NO
1722
1723  004350  005237  004430                INC   MEMCT
1724  004354  023727  004430  000003        CMP   MEMCT,#3
1725  004362  001404                        BEQ   15$
1726  004364  012737  052525  004434        MOV   #052525,PAT       ;DATA PATT FOR 2'ND HALF OF HI MEM
1727  004372  000734                        BR    10$               ;REPEAT
1728
1729  004374  012737  000006  000004  15$:  MOV   #ERRVEC+2,ERRVEC   ;RESTORE TIMEOUT VECTOR
1730  004402  005037  000006                CLR   ERRVEC+2
1731
1732  004406  032777  010000  174524        BIT   #BIT12,@SWR       ;SKIP TYPEOUT IF NOT SET
1733  004414  001412                        BEQ   CLKTST
1734  004416  104401  017607                TYPE  ,MEMORY
1735  004422  104401  017622                TYPE  ,DUN
1736  004426  000405                        BR    CLKTST
1737
1738  004430  000000          MEMCT:  0               ;0 = MEM FROM LSTAD TO MONITOR/ABS LOADER W/ 125252 PATT
1739                                                  ;1 = SAME WITH 052525 PATT
1740                                                  ;1 = MEM FROM 28K TO 30K WITH 125252 PAT
1741                                                  ;2 = SAME WITH 052525 PATT
1742                                                  ;3 = EXIT
1743  004432  000000          MEMHLD: 0               ;PUT MEMORY READ HERE
1744  004434  000000          PAT:    0               ;PATT TO BE WRITTEN & READ
1745  004436  000000          ADDR:   0               ;ADDR UNDER TEST
1746  004440  000000          HLD42:  0               ;SAVE LOC 42
1747
```

```
1748                                    ;;*****************************************************************
1749                                    ;       START LINE CLOCK
1750                                    ;;*****************************************************************
1751
1752  004442  032737  000200  001246  CLKTST: BIT     #BIT7,$DEVM     ;DROP TEST?
1753  004450  001031                          BNE     PRTST           ;BR IF YES
1754  004452  012737  012004  000100          MOV     #INTSRV,CLKVEC  ;SETUP VECTOR AREA
1755  004460  012737  000200  000102          MOV     #200,CLKVEC+2   ;LOCKOUT OTHER INTER.
1756  004466  005037  012012                  CLR     INTFLG
1757  004472  052737  000100  177546          BIS     #IE,CLK         ;SET CLOCK LOOSE!
1758
1759  004500  004537  011160                  JSR     R5,TIMER        ;SEE IF INTFLG GOES TO 100
1760  004504  012012                            INTFLG
1761  004506  000100                            100
1762  004510  104032                          ERROR   32              ;CLK NOT RESPONDING
1763  004512  000410                          BR      PRTST
1764
1765  004514  032777  010000  174416          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
1766  004522  001404                          BEQ     PRTST
1767  004524  104401  017110                  TYPE    ,CLOCK
1768  004530  104401  017534                  TYPE    ,RUN
```

```
1769
1770                            ;;*****************************************************************
1771                            ;       START PRINTER
1772                            ;;*****************************************************************
1773
1774  004534  032737  100000  001246  PRTST:  BIT     #BIT15,$DEVM     ;DROP TEST?
1775  004542  001074                   BNE     SMTST            ;BR IF YES
1776  004544  005737  010610           TST     PRPRES           ;PRINTER THERE?
1777  004550  001014                   BNE     1$               ;BR IF YES
1778
1779  004552  042737  000006  176610   BIC     #<RTS!DTR>,AMRC  ;MUST BE TOGGLED
1780  004560  052737  000006  176610   BIS     #<RTS!DTR>,AMRC  ;ELSE FORCE DTR IF LOOPBACK CONN INSTALLED
1781  004566  005737  177514           TST     PRS              ;NOW OK?
1782  004572  100460                   BMI     SMTST            ;EXIT IF DTR HAS NO EFFECT...NO LOOPBACK CONN.
1783  004574  005237  012146           INC     PRPORT           ;ELSE SET FLAG TO DO JUST 5 LINES
1784  004600  000402                   BR      2$
1785
1786  004602  005037  012146  1$:      CLR     PRPORT
1787  004606  012737  027014  177420  2$:      MOV     #<PRT!B9600!CHAR8>,PARAM        ;SETUP BAUD RATE & CHAR LENGTH
1788  004614  012737  012014  000200           MOV     #PRSRV,PRVEC     ;ELSE SETUP PRINTER VECTOR
1789  004622  012737  000200  000202           MOV     #200,PRVEC+2     ;LOCKOUT INTERR
1790
1791  004630  005037  012142                   CLR     LINCT            ;CLEAR LINE CTR
1792  004634  012737  000015  012140           MOV     #CR,PRCHR        ;CHAR TO BE PRINTED
1793                                                                    ;'DEL' (177) CAN BE USED TO CLR PRINTER BUFF
1794  004642  052737  000100  177514           BIS     #IE,PRS          ;SET PRINTER LOOSE!
1795                                                                    ;WILL PRINT CONTINUOUS 132 COLUMN LINES
1796                                                                    ;EA BEGINNING WITH ASCII 40 THRU 176
1797  004650  004537  011160                   JSR     R5,TIMER         ;SEE IF LINCT GOES TO 5
1798  004654  012142                             LINCT
1799  004656  000005                             5
1800  004660  104033                   ERROR   33               ;PRINTER NOT RESPONDING
1801  004662  000424                   BR      SMTST
1802
1803  004664  005737  012146           TST     PRPORT           ;5 LINES ONLY?
1804  004670  001403                   BEQ     3$               ;BR IF NO
1805  004672  042737  000100  177514   BIC     #IE,PRS          ;ELSE SHUT DOWN PRINTER
1806
1807  004700  032777  010000  174232  3$:      BIT     #BIT12,@SWR      ;SKIP TYPEOUT IF NOT SET
1808  004706  001412                   BEQ     SMTST
1809  004710  104401  017050           TYPE    ,PRINT
1810  004714  005737  012146           TST     PRPORT           ;5 LINES ONLY?
1811  004720  001403                   BEQ     4$               ;BR IF NO
1812  004722  104401  017061           TYPE    ,PORT
1813  004726  000402                   BR      SMTST
1814  004730  104401  017534          4$:      TYPE    ,RUN
```

```
1815
1816                                    ;;*****************************************************************
1817                                    ;       START SYNC COMM PORT
1818                                    ;;*****************************************************************
1819
1820  004734  032737  002000  001246  SMTST:  BIT     #BIT10,$DEVM    ;DROP TEST?
1821  004742  001402                          BEQ     20$             ;BR IF NO
1822  004744  000137  005222                  JMP     AMTST           ;ELSE JUMP TEST
1823
1824  004750  005037  005220          20$:    CLR     FREE
1825
1826  004754  005037  012636          8$:     CLR     LSTCHR          ;LAST CHAR FLAG FOR XMIT
1827  004760  052737  000400  176624          BIS     #MR,SMXC        ;MASTER RESET
1828  004766  052737  004000  176624          BIS     #MCLK,SMXC      ;MAINT CLOCK
1829
1830  004774  032737  000010  001246          BIT     #BIT3,$DEVM     ;EXT LOOPBACK?
1831  005002  001403                          BEQ     1$              ;BR IF YES
1832  005004  052737  014000  176624          BIS     #<MM!MCLK>,SMXC ;ELSE SET MAINT MODE FOR INT. LOOPBACK
1833  005012  012737  007026  176622  1$:     MOV     #<CHR8!ODDPAR!026>,SMPAR        ;SETUP SYNC PARAMETER REGISTER
1834  005020  052737  000026  176620          BIS     #<RTS!DTR!SRSYN>,SMRC   ;SET REQ TO SEND, DATA TERM RDY
1835                                                                          ;& SEARCH SYNC
1836  005026  052737  000020  176624          BIS     #SEND,SMXC      ;ENABLE XMIT SEND
1837  005034  012737  000026  176626          MOV     #026,SMXB       ;XMIT SYNC CHAR
1838
1839  005042  004537  011442                  JSR     R5,TIMER2       ;WAIT FOR DONE
1840  005046  176620                              SMRC
1841  005050  000200                              DONE
1842  005052  104045                          ERROR   45              ;NO DONE
1843  005054  000435                          BR      5$              ;EXIT
1844
1845  005056  013737  176622  012546          MOV     SMRB,COMHLD     ;READ WORD
1846  005064  023727  012546  000026          CMP     COMHLD,#026     ;IS IT SYNC CHAR?
1847  005072  001402                          BEQ     4$              ;BR IF YES
1848  005074  104006                          ERROR   6               ;DID NOT REC SYNC CHAR
1849  005076  000424                          BR      5$              ;EXIT
1850
1851  005100  004537  011134          4$:     JSR     R5,SETVEC       ;SETUP VECTOR AREA
1852  005104  000340                              SMRVEC
1853  005106  012640                              SMRSRV
1854  005110  012550                              SMXSRV
```

```
 1855
 1856  005112  012737  000027  012544          MOV     #027,CCMCHR     ;1'ST CHAR TO BE XMITTED
 1857  005120  052737  000100  176620          BIS     #IE,SMRC        ;SET SYNC MODEM LOOSE!
 1858  005126  052737  000100  176624          BIS     #IE,SMXC
 1859
 1860  005134  004537  011160                  JSR     R5,TIMER        ;SEE IF COMCHR GOES TO ALL 1'S (DONE)
 1861  005140  012544                            COMCHR
 1862  005142  177777                            177777
 1863  005144  000415                          BR      6$
 1864  005146  000240                          NOP
 1865
 1866  005150  012737  000400  176624  5$:     MOV     #MR,SMXC        ;MASTER RESET
 1867  005156  032777  010000  173754          BIT     #BIT12,@SWR     ;DO TYPEOUT IF SET
 1868  005164  001416                          BEQ     AMTST           ;EXIT IF NOT
 1869  005166  104401  017021                  TYPE    ,SCOM
 1870  005172  104401  017622                  TYPE    ,DUN
 1871  005176  000411                          BR      AMTST           ;EXIT
 1872
 1873  005200  005737  005220          6$:     TST     FREE            ;ALLOW 1 FREE ERROR FOR APT BREAK IN
 1874  005204  001402                          BEQ     7$
 1875  005206  104037                          ERROR   37              ;SYNC COMM NOT RESPONDING
 1876  005210  000757                          BR      5$
 1877
 1878  005212  005237  005220          7$:     INC     FREE
 1879  005216  000656                          BR      8$
 1880
 1881  005220  000000                  FREE:   0                       ;ERROR COUNTER
```

```
1882
1883                                    ;;*****************************************************************
1884                                    ;       START ASYNC COMM PORT
1885                                    ;;*****************************************************************
1886
1887  005222  032737  004000  001246  AMTST:  BIT     #BIT11,$DEVM    ;DROP TEST?
1888  005230  001054                          BNE     CL1TST          ;BR IF YES
1889
1890  005232  032737  000010  001246          BIT     #BIT3,$DEVM     ;EXT LOOPBACK?
1891  005240  001404                          BEQ     1$              ;BR IF YES
1892  005242  012737  037036  177420          MOV     #<AMOD!B9600!OPAR!CHAR8!MAINT>,PARAM    ;ELSE SET FOR INT LOOPBACK
1893  005250  000403                          BR      3$
1894  005252  012737  037034  177420  1$:     MOV     #<AMOD!B9600!OPAR!CHAR8>,PARAM  ;WAKE UP AMOD IF JUST DID SMOD
1895  005260  052737  000006  176610  3$:     BIS     #<RTS!DTR>,AMRC ;SET REQ TO SEND & DATA TERM RDY
1896
1897  005266  004537  011134                  JSR     R5,SETVEC       ;SETUP VECTOR AREA
1898  005272  000330                              AMRVEC
1899  005274  012466                              AMRSRV
1900  005276  012450                              AMXSRV
1901
1902  005300  005037  012544                  CLR     COMCHR          ;CHAR TO BE XMITTED
1903  005304  013737  176612  012546          MOV     AMRB,COMHLD     ;CLR OUT ANY PREVIOUS STORED WORD
1904  005312  052737  000100  176610          BIS     #IE,AMRC        ;SET ASYNC MODEM LOOSE!
1905  005320  052737  000100  176614          BIS     #IE,AMXC
1906
1907  005326  004537  011160                  JSR     R5,TIMER        ;SEE IF COMCHR GOES TO 377
1908  005332  012544                            COMCHR
1909  005334  000377                            377
1910  005336  104040                          ERROR   40              ;ASYNC COMM NOT RESPONDING
1911  005340  000410                          BR      CL1TST
1912
1913  005342  032777  010000  173570          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
1914  005350  001404                          BEQ     CL1TST
1915  005352  104401  017034                  TYPE    ,ACOM
1916  005356  104401  017534                  TYPE    ,RUN
```

```
                                ;;*****************************************************************
                                ;       START CLUSTER TERMINAL #1
                                ;;*****************************************************************

005362  005737  010606        CL1TST: TST     CLPRES          ;OPTION PRESENT?
005366  001002                        BNE     1$              ;BR IF YES
005370  000137  006036                JMP     DXTST0

005374  032737  004000  001246  1$:     BIT     #4000,$DEVM     ;DROP ASYNC?
005402  001406                        BEQ     4$              ;BR IF NO
005404  042737  000006  176610        BIC     #<RTS!DTR>,AMRC ;MUST BE TOGGLED
005412  052737  000006  176610        BIS     #<RTS!DTR>,AMRC ;PRIME IT

005420  032737  010000  001246  4$:     BIT     #BIT12,$DEVM    ;DROP TEST?
005426  001051                        BNE     CL2TST          ;BR IF YES

005430  032737  000001  001246        BIT     #BIT0,$DEVM     ;TERM #1 SET FOR EXT LOOPBACK?
005436  001404                        BEQ     2$              ;BR IF YES
005440  012737  005016  177420        MOV     #<CT1!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
005446  000403                        BR      3$
005450  012737  005014  177420  2$:     MOV     #<CT1!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK

005456  004537  011134          3$:     JSR     R5,SETVEC       ;SETUP INTERR VECTORS
005462  000300                            TK1VEC
005464  012166                            TK1SRV
005466  012150                            TP1SRV

005470  005037  012244                    CLR     T1CHR           ;CHAR TO BE XMITTED
005474  013737  176502  012246            MOV     TK1B,T1HLD      ;READ CHAR (CLEAR STALE DATA)
005502  052737  000100  176500            BIS     #IE,TK1S        ;SET CLUSTER TERM #1 LOOSE!
005510  052737  000100  176504            BIS     #IE,TP1S

005516  004537  011160                    JSR     R5,TIMER        ;SEE IF T1CHR GOES TO 377
005522  012244                              T1CHR
005524  000377                              377
005526  104034                            ERROR   34              ;CLUSTER TERM 1 NOT RESPONDING
005530  000410                            BR      CL2TST

005532  032777  010000  173400            BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
005540  001404                            BEQ     CL2TST
005542  104401  016766                    TYPE    ,CLT1
005546  104401  017534                    TYPE    ,RUN
```

```
1960
1961                                    ;;*****************************************************************
1962                                    ;       START CLUSTER TERMINAL #2
1963                                    ;;*****************************************************************
1964
1965  005552  032737  020000  001246  CL2TST: BIT     #BIT13,$DEVM    ;DROP TEST?
1966  005560  001051                          BNE     CL3TST          ;BR IF YES
1967
1968  005562  032737  000002  001246          BIT     #BIT1,$DEVM     ;TERM #2 SET FOR EXT LOOPBACK?
1969  005570  001404                          BEQ     1$              ;BR IF YES
1970  005572  012737  055016  177420          MOV     #<CT2!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
1971  005600  000403                          BR      2$
1972  005602  012737  055014  177420  1$:     MOV     #<CT2!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK
1973
1974  005610  004537  011134          2$:     JSR     R5,SETVEC       ;SETUP INTERR VECTORS
1975  005614  000310                              TK2VEC
1976  005616  012266                              TK2SRV
1977  005620  012250                              TP2SRV
1978
1979  005622  005037  012344                  CLR     T2CHR           ;CHAR TO BE XMITTED
1980  005626  013737  176512  012346          MOV     TK2B,T2HLD      ;READ CHAR (CLEAR STALE DATA)
1981  005634  052737  000100  176510          BIS     #IE,TK2S        ;SET CLUSTER TERM #2 LOOSE!
1982  005642  052737  000100  176514          BIS     #IE,TP2S
1983
1984  005650  004537  011160                  JSR     R5,TIMER        ;SEE IF T2CHR GOES TO 377
1985  005654  012344                            T2CHR
1986  005656  000377                            377
1987  005660  104035                          ERROR   35              ;CLUSTER TERM 2 NOT RESPONDING
1988  005662  000410                          BR      CL3TST
1989
1990  005664  032777  010000  173246          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
1991  005672  001404                          BEQ     CL3TST
1992  005674  104401  016777                  TYPE    ,CLT2
1993  005700  104401  017534                  TYPE    ,RUN
```

```
                                ;;*****************************************************************
                                ;       START CLUSTER TERMINAL #3
                                ;;*****************************************************************

005704  032737  040000  001246  CL3TST: BIT     #BIT14,$DEVM    ;DROP TEST?
005712  001051                          BNE     DXTST0          ;BR IF YES

005714  032737  000004  001246          BIT     #BIT2,$DEVM     ;TERM #3 SET FOR EXT LOOPBACK?
005722  001404                          BEQ     1$              ;BR IF YES
005724  012737  065016  177420          MOV     #<CT3!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
005732  000403                          BR      2$
005734  012737  065014  177420  1$:     MOV     #<CT3!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK

005742  004537  011134          2$:     JSR     R5,SETVEC       ;SETUP INTERR VECTORS
005746  000320                              TK3VEC
005750  012366                              TK3SRV
005752  012350                              TP3SRV

005754  005037  012444                  CLR     T3CHR           ;CHAR TO BE XMITTED
005760  013737  176522  012446          MOV     TK3B,T3HLD      ;READ CHAR (CLEAR STALE DATA)
005766  052737  000100  176520          BIS     #IE,TK3S        ;SET CLUSTER TERM #3 LOOSE!
005774  052737  000100  176524          BIS     #IE,TP3S

006002  004537  011160                  JSR     R5,TIMER        ;SEE IF T3CHR GOES TO 377
006006  012444                            T3CHR
006010  000377                            377
006012  104036                  ERROR   36              ;CLUSTER TERM 3 NOT RESPONDING
006014  000410                  BR      DXTST0

006016  032777  010000  173114  BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
006024  001404                  BEQ     DXTST0
006026  104401  017010          TYPE    ,CLT3
006032  104401  017534          TYPE    ,RUN
```

```
2028
2029                                    ;;*****************************************************************
2030                                    ;       FILL & EMPTY BUFFER TEST
2031                                    ;
2032                                    ;A FILL BUFFER COMMAND IS ISSUED WITH ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
2033                                    ;AN EMPTY BUFFER COMMAND IS ISSUED TO FILL 'DOBUF'.
2034                                    ;UPON COMPLETION, DOBUF IS CHECKED FOR ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
2035                                    ;ERRORS IN THIS TEST MAY INDICATE CABLE CROSS-TALK INTERFERENCE.
2036                                    ;;*****************************************************************
2037
2038  006036  005037  003120           DXTST0: CLR     DXTST           ;0 = DXTST0
2039
2040  006042  032737  000400  001246           BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2041  006050  001404                           BEQ     7$              ;BR IF NO
2042  006052  032737  001000  001246           BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2043  006060  001107                           BNE     DXTST1          ;BR IF YES
2044
2045  006062  004537  011442           7$:     JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2046  006066  177170                               RXCS
2047  006070  000200                               DONE
2048  006072  104075                           ERROR   75              ;NO DONE
2049  006074  000474                           BR      8$              ;EXIT
2050
2051  006076  012737  000011  177170           MOV     #INITAL,RXCS    ;DO AN INITIALIZE COMMAND
2052
2053  006104  004537  011442                   JSR     R5,TIMER2       ;WAIT FOR DONE
2054  006110  177170                               RXCS
2055  006112  000200                               DONE
2056  006114  104075                           ERROR   75              ;NO DONE
2057  006116  000463                           BR      8$
2058                                                                   ;FILL BUFFER
2059  006120  012700  000100                   MOV     #100,R0         ;WORD CT
2060  006124  005001                           CLR     R1              ;WORD
2061  006126  012737  000001  177170           MOV     #FBUF,RXCS      ;ISSUE FILL BUFF COMMAND
2062  006134  010137  177172           1$:     MOV     R1,RXDB
2063  006140  005101                           COM     R1              ;COMPLEMENT WORD
2064  006142  005300                           DEC     R0              ;DONE ALL 100 WORDS?
2065  006144  001373                           BNE     1$              ;BR IF NO
2066
2067  006146  004537  011442                   JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2068  006152  177170                               RXCS
2069  006154  000200                               DONE
2070  006156  104075                           ERROR   75              ;NO DONE
2071  006160  000442                           BR      8$              ;EXIT
2072
2073                                                                   ;EMPTY BUFFER
2074  006162  012700  000100                   MOV     #100,R0         ;WORD CT
2075  006166  012701  002424                   MOV     #DOBUF,R1       ;BUFFER ADDR
2076  006172  012737  000003  177170           MOV     #EBUF,RXCS      ;ISSUE EMPTY BUFF COMMAND
2077  006200  013721  177172           2$:     MOV     RXDB,(R1)+      ;STORE WORD
2078  006204  005300                           DEC     R0              ;DONE?
2079  006206  001374                           BNE     2$              ;BR IF NO
```

```
 2080
 2081  006210  004537  011442            JSR     R5,TIMER2       ;WAIT FOR DISK DONE
 2082  006214  177170                       RXCS
 2083  006216  000200                       DONE
 2084  006220  104075                    ERROR   75              ;NO DONE
 2085  006222  000421                    BR      8$              ;EXIT
 2086
 2087                                                            ;CHECK IT
 2088  006224  012701  002424            MOV     #DOBUF,R1       ;BUFFER ADDR
 2089  006230  012137  006272    3$:     MOV     (R1)+,EBHLD     ;STORE IT
 2090  006234  001401                    BEQ     4$              ;BR IF ZERO
 2091  006236  104076                    ERROR   76              ;WORD NOT ALL ZEROS
 2092
 2093  006240  012137  006272    4$:     MOV     (R1)+,EBHLD
 2094  006244  023727  006272  177777    CMP     EBHLD,#-1       ;ALL 1'S?
 2095  006252  001401                    BEQ     5$              ;BR IF YES
 2096  006254  104077                    ERROR   77              ;WORD NOT ALL ONES
 2097
 2098  006256  020127  002624    5$:     CMP     R1,#DOBUF+200   ;END OF DOBUFF?
 2099  006262  001362                    BNE     3$              ;BR IF NO
 2100  006264  000405                    BR      DXTST1          ;GO TO NXT TST
 2101
 2102  006266  000137  010114    8$:     JMP     EOP             ;EXIT ALL DISK TESTS
 2103
 2104  006272  000000            EBHLD:  0                       ;STORAGE        FOR ERROR REPORT
 2105  006274  000000            EXP0:   0                       ;EXPECT 0'S     FOR ERROR REPORT
 2106  006276  177777            EXP1:   -1                      ;EXPECT 1'S     FOR ERROR REPORT
```

```
2107
2108                                    ;;*****************************************************************
2109                                    ;       START DX0 & DX1 DATA PATTERN
2110                                    ;
2111                                    ;EACH SECTOR IS FILLED WITH ITS TRACK & SECTOR ADDRESS AS DATA.
2112                                    ;DX1 WILL ALWAYS HAVE ITS DATA BIT 15 SET TO DISTINGUISH IT FROM DX0.
2113                                    ;EACH TRACK IS READ & CHECKED BEFORE GOING TO THE NEXT TRACK.
2114                                    ;DATA IS WRITTEN ON ALTERANTE SECTORS TO AVOID LATENCY TIMES.
2115                                    ;IE: SECTORS 1,3,5...31 FOLLOWED BY 2,4,6...32.
2116                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2117                                    ;1'ST PASS PERFORMED ON 1'ST 10 TRACKS. SUBSEQUENT PASSES ON ALL TRACKS
2118                                    ;;*****************************************************************
2119
2120  006300  012737  000001  003120  DXTST1: MOV     #1,DXTST        ;1 = DATA PATT TESTS
2121
2122  006306  005037  002400                CLR     D0PAT           ;INIT DX0 FLAGS
2123  006312  005037  002404                CLR     D0ERR
2124  006316  005037  002414                CLR     D0CERR
2125  006322  005037  002406                CLR     D0TRK
2126  006326  012737  000001  002410        MOV     #1,D0SEC
2127
2128  006334  005037  002650                CLR     D1PAT           ;INIT DX1 FLAGS
2129  006340  005037  002654                CLR     D1ERR
2130  006344  005037  002664                CLR     D1CERR
2131  006350  005037  002656                CLR     D1TRK
2132  006354  012737  000001  002660        MOV     #1,D1SEC
2133
2134  006362  032737  000400  001246        BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2135  006370  001031                        BNE     2$              ;BR IF YES
2136
2137  006372  012737  012740  000264        MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2138  006400  012737  000200  000266        MOV     #200,RXVEC+2
2139
2140  006406  005737  003134        1$:     TST     COMP2           ;DOING COMPAT PASS 2?
2141  006412  001404                        BEQ     10$             ;BR IF NO
2142  006414  012737  013432  013040        MOV     #DORSEC,DODISP  ;ELSE DO READS ONLY
2143  006422  000403                        BR      7$
2144
2145  006424  012737  013056  013040  10$:  MOV     #DOFBUF,DODISP  ;DO NORMAL TESTING
2146  006432  005037  003122        7$:     CLR     FIN             ;DO DX1 WHEN SET
2147  006436  052737  000100  177170        BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE!
2148
2149  006444  004737  011244                JSR     PC,TIMDO
2150  006450  104041                        ERROR   41              ;DX0 NOT GETTING TRACK DONE FLAG
2151  006452  000240                        NOP
```

```
2152
2153  006454  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2154  006462  001031                          BNE     4$              ;BR IF YES
2155
2156  006464  012737  012740  000264          MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2157  006472  012737  000200  000266          MOV     #200,RXVEC+2
2158
2159  006500  005737  003134          3$:     TST     COMP2           ;DOING COMPAT PASS 2?
2160  006504  001404                          BEQ     11$             ;BR IF NO
2161  006506  012737  015216  013046          MOV     #D1RSEC,D1DISP  ;ELSE DO READ ONLY
2162  006514  000403                          BR      8$
2163
2164  006516  012737  014642  013046  11$:     MOV     #D1FBUF,D1DISP  ;DO NORMAL TESTING
2165  006524  005037  003122          8$:     CLR     FIN             ;GO BACK TO DX0 WHEN SET
2166  006530  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2167
2168  006536  004737  011340                  JSR     PC,TIMD1
2169  006542  104042                          ERROR   42              ;DX1 NOT GETTING TRACK DONE FLAG
2170  006544  000240                          NOP
2171
2172  006546  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2173  006554  001007                          BNE     5$              ;BR IF YES
2174  006556  005737  002400                  TST     DOPAT           ;ELSE IS DX0 ALL DONE?
2175  006562  001004                          BNE     5$              ;BR IF YES
2176  006564  005737  011336                  TST     DOTMO           ;TIMEOUT?
2177  006570  001706                          BEQ     1$              ;BR IF NO
2178  006572  000717                          BR      7$              ;ELSE CONT LAST COMMAND
2179
2180  006574  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2181  006602  001007                          BNE     6$              ;BR IF YES
2182  006604  005737  002650                  TST     D1PAT           ;ELSE IS DX1 ALL DONE?
2183  006610  001004                          BNE     6$              ;BR IF YES
2184  006612  005737  011426                  TST     D1TMO           ;TIMEOUT?
2185  006616  001730                          BEQ     3$              ;BR IF NO
2186  006620  000741                          BR      8$              ;ELSE CONT LAST COMMAND
2187
2188  006622  005737  003132          6$:     TST     COMP1           ;DOING COMPAT PASS 1?
2189  006626  001402                          BEQ     DXTST2          ;BR IF NO
2190  006630  000137  010114                  JMP     EOP             ;ELSE ALL DONE
2191
```

```
                                ;;*****************************************************************
                                ;       START DX0 & DX1 RANDOM SEEKS
                                ;
                                ;RANDOM SEEKS ARE PERFORMED ON EACH DISK.
                                ;DATA IS READ & VERIFIED TO BE CORRECT FROM THE PREVIOUS TEST.
                                ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
                                ;1'ST PASS PERFORMS 20 RANDOM SEEKS ON 1'ST 10 TRACKS.
                                ;SUBSEQUENT PASSES PERFORMS 500 RANDOM SEEKS BETWEEN ALL TRACKS.
                                ;;*****************************************************************

006634  012737  000002  003120  DXTST2: MOV     #2,DXTST        ;2 = RANDOM SEEK TESTS

006642  005037  002404                  CLR     D0ERR           ;INIT DX0 FLAGS
006646  005037  002414                  CLR     D0CERR
006652  005037  002402                  CLR     D0CNT

006656  005037  002654                  CLR     D1ERR           ;INIT DX1 FLAGS
006662  005037  002664                  CLR     D1CERR
006666  005037  002652                  CLR     D1CNT

006672  032737  000400  001246          BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
006700  001041                          BNE     2$              ;BR IF YES

006702  005037  010734          1$:     CLR     LOLIM
006706  013737  003124  010736          MOV     MAXTRK,HILIM
006714  004737  010632                  JSR     PC,RAND
006720  010137  002406                  MOV     R1,D0TRK        ;GET RANDOM TRACK #

006724  005237  010734                  INC     LOLIM
006730  012737  000032  010736          MOV     #32,HILIM
006736  004737  010632                  JSR     PC,RAND
006742  010137  002410                  MOV     R1,D0SEC        ;GET RANDOM SECTOR #

006746  004537  011616                  JSR     R5,CK0BAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
006752  000753                              BR  1$              ;BR IF RETURN HERE

006754  012737  013432  013040          MOV     #D0RSEC,D0DISP  ;SET DISPATCH TABLE TO POINT TO
                                                                ;READ SECTOR HANDLER.
006762  005037  003122                  CLR     FIN             ;DO DX1 WHEN SET
006766  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE!

006774  004737  011244                  JSR     PC,TIMD0
007000  104041                          ERROR   41              ;DX0 NOT GETTING DONE FLAG
007002  000240                          NOP
```

```
2237
2238  007004  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2239  007012  001041                          BNE     4$              ;BR IF YES
2240
2241  007014  005037  010734          3$:     CLR     LOLIM
2242  007020  013737  003124  010736          MOV     MAXTRK,HILIM
2243  007026  004737  010632                  JSR     PC,RAND
2244  007032  010137  002656                  MOV     R1,D1TRK        ;GET RANDOM TRACK #
2245
2246  007036  005237  010734                  INC     LOLIM
2247  007042  012737  000032  010736          MOV     #32,HILIM
2248  007050  004737  010632                  JSR     PC,RAND
2249  007054  010137  002660                  MOV     R1,D1SEC        ;GET RANDOM SECTOR #
2250
2251  007060  004537  011730                  JSR     R5,CK1BAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
2252  007064  000753                          BR      3$              ;BR IF RETURN HERE
2253
2254  007066  012737  015216  013046          MOV     #D1RSEC,D1DISP  ;SET DISPATCH TABLE TO POINT TO
2255                                                                  ;READ SECTOR HANDLER.
2256  007074  005037  003122                  CLR     FIN             ;GO BACK TO DX0 WHEN SET
2257  007100  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2258
2259  007106  004737  011340                  JSR     PC,TIMD1
2260  007112  104042                          ERROR   42              ;DX1 NOT GETTING DONE FLAG
2261  007114  000240                          NOP
2262
2263  007116  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2264  007124  001004                          BNE     5$              ;BR IF YES
2265  007126  023737  002402  003126          CMP     D0CNT,MAXSK     ;DID ALL SEEKS?
2266  007134  001262                          BNE     1$              ;BR IF NO
2267
2268  007136  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2269  007144  001004                          BNE     6$              ;BR IF YES
2270  007146  023737  002652  003126          CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
2271  007154  001317                          BNE     3$              ;BR IF NO
2272
2273  007156  005737  003134          6$:     TST     COMP2           ;DOING COMPAT PASS 2?
2274  007162  001402                          BEQ     DXTST3          ;BR IF NO
2275  007164  000137  010114                  JMP     EOP             ;ELSE ALL DONE
2276
```

```
                                ;;*****************************************************************
                                ;       INITIALIZE COMMAND TEST
                                ;
                                ;THE INITIALIZE  & EMPTY BUFFER COMMAND IS ISSUED TO DX0.
                                ;DATA IS VERIFIED TO BE THAT OF TRACK 1, SECTOR 1.
                                ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
                                ; 'INIT' CANNOT BE ISSUED TO DX1.
                                ;;*****************************************************************

007170  032737  000400  001246  DXTST3: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
007176  001062                        BNE     DXTST4          ;BR IF YES

007200  012737  000003  003120          MOV     #3,DXTST        ;3 = INIT TEST
007206  005037  002404                  CLR     DOERR           ;FLAGS
007212  005037  002414                  CLR     DOCERR

007216  012737  014476  013040          MOV     #DOINIT,DODISP  ;SET DISPATCH TABLE TO GO TO INIT HANDLER

007224  005037  003122                  CLR     FIN
007230  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

007236  004737  011244                  JSR     PC,TIMDO
007242  104046                          ERROR   46              ;DX0 HUNG ON INITIALIZE COMMAND
007244  000437                          BR      DXTST4

007246  005737  177170                  TST     RXCS            ;ERROR?
007252  100001                          BPL     1$              ;BR IF NO
007254  104063                          ERROR   63              ;INIT COMMAND ERROR

007256  032737  000001  177172  1$:     BIT     #ID,RXES        ;INIT DONE BIT SET?
007264  001001                          BNE     2$              ;BR IF YES
007266  104055                          ERROR   55              ;INIT DONE NOT SET

007270  005037  002404          2$:     CLR     DOERR
007274  005037  002414                  CLR     DOCERR
007300  012737  000001  002406          MOV     #1,DOTRK        ;EXP TRK & SEC
007306  012737  000001  002410          MOV     #1,DOSEC

007314  012737  013712  013040          MOV     #DOEBUF,DODISP  ;SETUP DISPATCH TABLE TO
                                                                ;EMPTY BUFFER & VERIFY DATA
007322  005037  003122                  CLR     FIN
007326  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

007334  004737  011244                  JSR     PC,TIMDO
007340  104056                          ERROR   56              ;DX0 HUNG ON EMPTY BUFF COMMAND
007342  000240                          NOP
```

```
2324
2325                                    ;;*****************************************************************
2326                                    ;       RESTORE COMMAND TEST
2327                                    ;
2328                                    ;THIS TEST VERIFIES THAT THE RESTORE COMMAND CAN BE ISSUED
2329                                    ;TO DX0 & DX1 & THAT NO ERRORS RESULT FROM IT.
2330                                    ;;*****************************************************************
2331
2332  007344  032737  000400  001246  DXTST4: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2333  007352  001121                          BNE     DXTST5          ;BR IF YES
2334
2335  007354  012737  000004  003120          MOV     #4,DXTST        ;4 = RESTORE TEST
2336  007362  012737  014522  013040          MOV     #DOREST,DODISP  ;SETUP DISPATCH TABLE
2337  007370  005037  003122                  CLR     FIN
2338  007374  052737  000100  177170          BIS     #IE,RXCS
2339
2340  007402  004737  011244                  JSR     PC,TIMD0
2341  007406  104047                          ERROR   47              ;DX0 HUNG ON RESTORE COMMAND
2342  007410  000431                          BR      2$
2343
2344  007412  005737  177170                  TST     RXCS            ;ERROR?
2345  007416  100001                          BPL     1$              ;BR IF NO
2346  007420  104054                          ERROR   54              ;DX0 RESTORE COMMAND ERROR
2347
2348  007422  005037  002406          1$:     CLR     DOTRK           ;CHK FOR RECORD NOT FOUND (RNF)
2349  007426  012737  000001  002410          MOV     #1,DOSEC        ;SET TRK/SEC. SHOULD NOT MOVE
2350                                                                  ;LOGIC THINKS ITS ALREADY THERE.
2351  007434  005037  002404                  CLR     DOERR
2352  007440  005037  002414                  CLR     DOCERR
2353  007444  005037  003122                  CLR     FIN
2354  007450  012737  013432  013040          MOV     #DORSEC,DODISP  ;SETUP TO READ SECTOR
2355  007456  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERR LOOSE
2356
2357  007464  004737  011244                  JSR     PC,TIMD0
2358  007470  104041                          ERROR   41              ;DX0 HUNG ON READ CMD
2359  007472  000240                          NOP
2360
2361  007474  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2362  007502  001045                          BNE     DXTST5          ;BR IF YES
2363
2364  007504  012737  014540  013046          MOV     #D1REST,D1DISP  ;REPEAT FOR DX1
2365  007512  005037  003122                  CLR     FIN
2366  007516  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE
2367
2368  007524  004737  011340                  JSR     PC,TIMD1
2369  007530  104042                          ERROR   42              ;DX1 HUNG ON RESTORE CMD
2370  007532  000431                          BR      DXTST5
2371
2372  007534  005737  177170                  TST     RXCS            ;ERROR?
2373  007540  100001                          BPL     3$              ;BR IF NO
2374  007542  104074                          ERROR   74              ;DX1 RESTORE CMD ERROR
```

```
2375
2376  007544  005037  002656          3$:     CLR     D1TRK           ;READ TO CHECK FOR RNF
2377  007550  012737  000001  002660          MOV     #1,D1SEC
2378  007556  005037  002654                  CLR     D1ERR
2379  007562  005037  002664                  CLR     D1CERR
2380  007566  005037  003122                  CLR     FIN
2381  007572  012737  015216  013046          MOV     #D1RSEC,D1DISP  ;SETUP TO READ SECTOR
2382  007600  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE
2383
2384  007606  004737  011340                  JSR     PC,TIMD1
2385  007612  104042                          ERROR   42              ;DX1 HUNG ON READ CMD
2386  007614  000240                          NOP
2387
2388
2389
2390                                  ;;*****************************************************************
2391                                  ;       WRITE SECTOR WITH DELETED DATA MARK TEST
2392                                  ;
2393                                  ;THIS TEST VERIFIES THAT THE WRITE DELETED DATA COMMAND CAN BE ISSUED
2394                                  ;& THAT THE 'DELETED DATA' BIT 5 IS SET IN RXCS AFTER READY IS RECV'D.
2395                                  ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2396                                  ;;*****************************************************************
2397
2398  007616  032737  000400  001246  DXTST5: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2399  007624  001043                          BNE     DXTST6          ;BR IF YES
2400
2401  007626  012737  000005  003120          MOV     #5,DXTST        ;5 = WR DEL DATA TEST
2402  007634  005037  002404                  CLR     D0ERR
2403  007640  005037  002414                  CLR     D0CERR
2404
2405  007644  012737  013056  013040          MOV     #D0FBUF,D0DISP  ;SETUP DISPATCH TABLE
2406
2407  007652  005037  003122                  CLR     FIN
2408  007656  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2409
2410  007664  004737  011244                  JSR     PC,TIMD0
2411  007670  104050                          ERROR   50              ;DX0 HUNG ON WRITE DEL DATA CMD
2412  007672  000420                          BR      DXTST6
2413
2414  007674  012737  014556  013040          MOV     #D0STAT,D0DISP  ;SETUP DISPATCH TABLE
2415
2416  007702  005037  003122                  CLR     FIN
2417  007706  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2418
2419  007714  004737  011244                  JSR     PC,TIMD0
2420  007720  104051                          ERROR   51              ;DX0 HUNG ON READ STATUS
2421  007722  000404                          BR      DXTST6
2422
2423  007724  005737  177170                  TST     RXCS            ;ERROR?
2424  007730  100001                          BPL     .+4             ;BR IF NO
2425  007732  104057                          ERROR   57              ;READ STATUS ERROR
```

```
                        ;;*****************************************************************
                        ;       INVALID ADDRESS TESTS
                        ;
                        ;A WRITE SECTOR COMMAND IS ISSUED TO INVALID TRACK 115(8)
                        ;& RXCS IS CHECKED FOR THE INVALID ADDR BIT 1 TO BE SET AFTER RDY.
                        ;THE ABOVE IS REPEATED FOR INVALID SECTOR 33(8).
                        ;;*****************************************************************
007734  032737  000400  001246  DXTST6: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
007742  001062                          BNE     4$              ;BR IF YES

007744  005037  010112                  CLR     INVCT
007750  012737  000006  003120          MOV     #6,DXTST        ;6 = INV ADDR TESTS
007756  012737  000115  002406          MOV     #115,D0TRK
007764  012737  000001  002410          MOV     #1,D0SEC
007772  012737  046401  177174          MOV     #46401,RXSA     ;INVALID TRACK

010000  012737  014574  013040  1$:     MOV     #D0INV,D0DISP   ;SETUP DISPATCH TABLE

010006  005037  003122                  CLR     FIN
010012  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

010020  004737  011244                  JSR     PC,TIMD0
010024  104060                          ERROR   60              ;DX0 HUNG ON INVALID ADDR
010026  000432                          BR      EOP

010030  032737  000002  177172          BIT     #INVADR,RXES    ;INV ADDR BIT SET?
010036  001002                          BNE     2$              ;BR IF YES
010040  104061                          ERROR   61              ;INVALID ADDR BIT NOT SET
010042  000404                          BR      3$

010044  005737  177170          2$:     TST     RXCS            ;ERROR BIT SET?
010050  100401                          BMI     3$              ;BR IF YES
010052  104062                          ERROR   62              ;ERROR BIT NOT SET

010054  005737  010112          3$:     TST     INVCT           ;DID WE JUST DO INV SECTOR?
010060  001013                          BNE     4$              ;BR IF YES...DONE

010062  005237  010112                  INC     INVCT
010066  005037  002406                  CLR     D0TRK
010072  012737  000033  002410          MOV     #33,D0SEC
010100  012737  000033  177174          MOV     #33,RXSA        ;INVALID SECTOR
010106  000734                          BR      1$              ;REPEAT FOR INVALID SECTOR

010110  000401                  4$:     BR      .+4

010112  000000                  INVCT:  0                       ;0 = DOING INVALID TRACK
                                                                ;1 = DOING INVALID SECTOR
```

```
2476                                    ;;*****************************************************************
2477                                    ;       END OF PASS
2478                                    ;;*****************************************************************
2479
2480  010114  005237  001176      EOP:  INC     $PASS           ;PASS CTR
2481  010120  104401  017630            TYPE    ,ENDPAS
2482  010124  013746  001176            MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
2483  010130  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2484  010132  104401  017645            TYPE    ,MSG1
2485  010136  013746  026052            MOV     TERR,-(SP)      ;;SAVE TERR FOR TYPEOUT
2486  010142  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2487  010144  104401  017674            TYPE    ,MSG2
2488  010150  013746  026054            MOV     TSERR,-(SP)     ;;SAVE TSERR FOR TYPEOUT
2489  010154  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2490  010156  104401  017730            TYPE    ,MSG3
2491  010162  013746  026056            MOV     PERR,-(SP)      ;;SAVE PERR FOR TYPEOUT
2492  010166  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2493  010170  104401  017765            TYPE    ,MSG4
2494  010174  013746  026060            MOV     PSERR,-(SP)     ;;SAVE PSERR FOR TYPEOUT
2495  010200  104405                    TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2496  010202  104401  010252            TYPE    ,NULL           ;NULL CHAR
2497  010206  012737  000114  003124    MOV     #114.,MAXTRK    ;FOR ALL SUBSEQUENT PASSES
2498  010214  012737  000764  003126    MOV     #500.,MAXSK
2499
2500  010222  005737  003132            TST     COMP1           ;COMPAT PASS 1?
2501  010226  001407                    BEQ     1$              ;BR IF NO
2502  010230  104401  020214            TYPE    ,COMPAT         ;DONE MSG
2503  010234  000000                    HALT
2504  010236  005037  003132            CLR     COMP1
2505  010242  005237  003134            INC     COMP2
2506  010246  000137  003706      1$:   JMP     LOOP            ;REPEAT
2507
2508  010252    377     377   000 NULL: .BYTE   -1,-1,0         ;NULL CHAR STRING
2509          010256                    .EVEN
2510
```

```
2511
2512                                    .SBTTL PROGRAM SUBROUTINES
2513
2514
2515  010256  012737  012004  000004  SIZE:   MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
2516  010264  012737  000200  000006          MOV     #200,ERRVEC+2   ;LOCKOUT INTERR
2517  010272  005037  012012                  CLR     INTFLG
2518
2519  010276  012700  017776                  MOV     #17776,R0       ;SIZE MEM. START WITH 4K
2520  010302  005001                          CLR     R1              ;MEM MAP, 1=4K, 2=8K, ETC
2521  010304  005037  012012                  CLR     INTFLG
2522
2523  010310  005710                  1$:     TST     (R0)
2524  010312  005737  012012                  TST     INTFLG          ;GOT TIMEOUT INTERR?
2525  010316  001007                          BNE     2$              ;BR IF YES
2526  010320  005201                          INC     R1
2527  010322  020027  157776                  CMP     R0,#157776      ;DONE 28K MEM?
2528  010326  001403                          BEQ     2$
2529  010330  062700  020000                  ADD     #20000,R0       ;GO TO NEXT 4K
2530  010334  000765                          BR      1$
2531
2532  010336  010037  010630          2$:     MOV     R0,MAXMEM       ;SAVE
2533  010342  162737  020000  010630          SUB     #20000,MAXMEM   ;GO BACK TO LAST VALID BLOCK
2534  010350  006301                          ASL     R1              ;MULT BY 2
2535  010352  016137  010610  010362          MOV     MEMTBL-2(R1),3$
2536  010360  104401                          TYPE
2537  010362  000000                  3$:     .WORD   0               ;MEM SIZE
2538  010364  104401  017160                  TYPE    ,MEM
2539
2540  010370  005000                          CLR     R0
2541  010372  005037  012012                  CLR     INTFLG
2542  010376  012700  176500                  MOV     #TK1S,R0        ;SETUP CLUSTER CSR ADDR
2543  010402  005710                  4$:     TST     (R0)            ;DARE IT TO TIMEOUT
2544  010404  005737  012012                  TST     INTFLG          ;DID IT?
2545  010410  001011                          BNE     6$              ;BR IF YES
2546  010412  020027  176520                  CMP     R0,#TK3S        ;DID ALL 3?
2547  010416  001403                          BEQ     5$              ;BR IF YES
2548  010420  062700  000010                  ADD     #10,R0          ;ELSE DO NEXT
2549  010424  000766                          BR      4$
2550
2551  010426  005237  010606          5$:     INC     CLPRES          ;OPTION PRESENT
2552  010432  000406                          BR      7$              ;ALL 3 MUST RESPOND
2553
2554  010434  005037  010606          6$:     CLR     CLPRES          ;OPTION NOT PRESENT
2555  010440  104401  016746                  TYPE    ,CLOPT
2556  010444  104401  017200                  TYPE    ,NOTPRES
2557
2558  010450  012737  000006  000004  7$:     MOV     #ERRVEC+2,ERRVEC        ;RESET TMO VECTOR
2559  010456  005037  000006                  CLR     ERRVEC+2
```

```
2560
2561  010462  005000                        CLR     R0
2562  010464  012701  000200                MOV     #BIT7,R1
2563
2564  010470  030137  001246        8$:     BIT     R1,$DEVM        ;DEVICE DROPPED?
2565  010474  001006                        BNE     10$             ;BR IF YES
2566  010476  000241                9$:     CLC
2567  010500  006301                        ASL     R1
2568  010502  103424                        BCS     13$             ;BR IF ALL BITS TESTED
2569  010504  062700  000002                ADD     #2,R0           ;ELSE BUMP INDEX
2570  010510  000767                        BR      8$
2571
2572  010512  016037  010532  010522  10$:  MOV     12$(R0),11$
2573  010520  104401                        TYPE
2574  010522  000000                11$:    .WORD   0               ;DEVICE TO BE DROPPED
2575  010524  104401  017513                TYPE    ,DROP
2576  010530  000762                        BR      9$
2577
2578  010532  017110                12$:    CLOCK
2579  010534  017076                        DRV0
2580  010536  017103                        DRV1
2581  010540  017021                        SCOM
2582  010542  017034                        ACOM
2583  010544  016766                        CLT1
2584  010546  016777                        CLT2
2585  010550  017010                        CLT3
2586  010552  017050                        PRINT
2587
2588  010554  005737  177514        13$:    TST     PRS             ;SEE IF PRINTER ERROR
2589  010560  100007                        BPL     14$             ;BR IF NO
2590  010562  005037  010610                CLR     PRPRES          ;CLR    FLAG
2591  010566  104401  017050                TYPE    ,PRINT          ;PRINTER NOT PRESENT
2592  010572  104401  017200                TYPE    ,NOTPRES
2593  010576  000402                        BR      15$
2594  010600  005237  010610        14$:    INC     PRPRES
2595
2596  010604  000207                15$:    RTS     PC
2597
2598
2599  010606  000000                CLPRES: 0                       ;CLUSTER PRESENT = NON-ZERO
2600  010610  000000                PRPRES: 0                       ;PRINTER PRESENT = NON-ZERO
2601
2602  010612  017117                MEMTBL: M4K                     ;MSG ADDRESSES
2603  010614  017123                        M8K
2604  010616  017127                        M12K
2605  010620  017134                        M16K
2606  010622  017141                        M20K
2607  010624  017146                        M24K
2608  010626  017153                        M30K
2609
2610  010630  000000                MAXMEM: 0                       ;MAX MEMORY
2611
2612
2613
```

```
                        ;;*****************************************************************
                        ;       RANDOM NUMBER GENERATOR
                        ;;*****************************************************************

010632  013700  010742  RAND:   MOV     LONUM,R0
010636  013701  010740          MOV     HINUM,R1
010642  012702  177771          MOV     #-7,R2
010646  006300          1$:     ASL     R0
010650  006101                  ROL     R1              ;ROTATE CARRY TO R1
010652  005202                  INC     R2              ;DONE?
010654  001374                  BNE     1$              ;BR IN NO
010656  063700  010742          ADD     LONUM,R0        ;ADD NUMBER TO MAKE X 129
010662  005501                  ADC     R1
010664  063701  010740          ADD     HINUM,R1        ;ADD NUMBER TO MAKE X 129
010670  062700  001057          ADD     #1057,R0        ;ADD LO CONSTANT
010674  005501                  ADC     R1
010676  062701  047401          ADD     #47401,R1       ;ADD HI CONSTANT
010702  010037  010742          MOV     R0,LONUM
010706  010137  010740          MOV     R1,HINUM
                                                        ;SCALE TO BE WITHIN LIMITS
010712  042701  177400          BIC     #177400,R1      ;SAVE LO BYTE ONLY
010716  020137  010734          CMP     R1,LOLIM
010722  103743                  BLO     RAND            ;BR IF N < LOLIM
010724  020137  010736          CMP     R1,HILIM
010730  101340                  BHI     RAND            ;BR IF N > HILIM
010732  000207                  RTS     PC              ;ELSE EXIT WITH R1 = N

010734  000000          LOLIM:  0
010736  000000          HILIM:  0

010740  176543          HINUM:  .WORD 176543
010742  123456          LONUM:  .WORD 123456


010744  123727  001210  000001  EXAMKB: CMPB    $ENV,#1         ;APT?
010752  001401                  BEQ     1$              ;BR IF YES
010754  104407                  CKSWR
010756  000207          1$:     RTS     PC
```

```
2652
2653                                    ;;*****************************************************************
2654                                    ;ROUTINE TO DISPLAY CURRENT DEVM & ALLOW OPERATOR TO INPUT NEW VALUE.
2655                                    ;;*****************************************************************
2656
2657  010760  104401  020311          GT$DEV: TYPE    ,DEVM           ;ELSE SHOW CURRENT
2658  010764  013746  001246                  MOV     $DEVM,-(SP)     ;;SAVE $DEVM FOR TYPEOUT
2659  010770  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2660  010772  104401  025222                  TYPE    ,$MNEW          ;GET NEW
2661
2662  010776  005046                  1$:     CLR     -(SP)           ;CLR CTR
2663  011000  005046                          CLR     -(SP)           ;CLR NEW DEVM
2664
2665  011002  105777  170136          2$:     TSTB    @$TKS           ;CHAR THERE?
2666  011006  100375                          BPL     2$              ;BR IF NO
2667  011010  117746  170132                  MOVB    @$TKB,-(SP)     ;ELSE GET CHAR
2668  011014  042716  177600                  BIC     #^C177,(SP)     ;MAKE IT 7 BIT ASCII
2669
2670  011020  021627  000015          3$:     CMP     (SP),#15        ;<CR>?
2671  011024  001013                          BNE     7$              ;BR IF NO
2672  011026  005766  000004                  TST     4(SP)           ;ELSE IS IT 1'ST CHAR?
2673  011032  001403                          BEQ     4$              ;BR IF YES
2674  011034  016637  000002  001246          MOV     2(SP),$DEVM     ;ELSE SAVE NEW DEVM
2675  011042  062706  000006          4$:     ADD     #6,SP           ;RESTORE STACK
2676  011046  104401  001165          5$:     TYPE    ,$CRLF
2677  011052  000002                  6$:     RTI
2678
2679  011054  004737  024322          7$:     JSR     PC,$TYPEC       ;ECHO CHAR
2680  011060  021627  000060                  CMP     (SP),#60        ;CHAR < 0?
2681  011064  002420                          BLT     9$              ;BR IF YES
2682  011066  021627  000067                  CMP     (SP),#67        ;CHAR >7?
2683  011072  003015                          BGT     9$              ;BR IF YES
2684  011074  042726  000060                  BIC     #60,(SP)+       ;STRIP OFF ASCII
2685  011100  005766  000002                  TST     2(SP)           ;IS IT 1'ST CHAR?
2686  011104  001403                          BEQ     8$              ;BR IF YES
2687  011106  006316                          ASL     (SP)            ;ELSE SHIFT PRESENT CHAR
2688  011110  006316                          ASL     (SP)            ;TO MAKE ROOM
2689  011112  006316                          ASL     (SP)            ;FOR NEW ONE
2690
2691  011114  005266  000002          8$:     INC     2(SP)           ;KEEP CHAR CT
2692  011120  056616  177776                  BIS     -2(SP),(SP)     ;SET IN NEW CHAR
2693  011124  000726                          BR      2$              ;GET NEW ONE
2694
2695  011126  104401  001164          9$:     TYPE    ,$QUES          ;TYPE ?<CRLF>
2696  011132  000747                          BR      6$              ;DO ALL OVER
2697
```

```
                        ;;*****************************************************************
                        ;* ROUTINE TO SET 2 CONSECUTIVE INTERRUPT VECTORS
                        ;* 3 ARGUMENTS ARE PASSED THRU R5:
                        ;*      VECTOR ADDRESS
                        ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR RECVR
                        ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR XMITTER.
                        ;* PRIORITY WILL BE SET TO DISABLE FURTHUR INTERR.
                        ;;*****************************************************************

011134  010046          SETVEC: MOV     R0,-(SP)                ;SAVE
011136  012500                  MOV     (R5)+,R0                ;GET VECTOR ADDR
011140  012520                  MOV     (R5)+,(R0)+             ;PUT SERV ROUTINE ADDR THERE
011142  012720  000200          MOV     #200,(R0)+              ;DISABLE INTERR
011146  012520                  MOV     (R5)+,(R0)+             ;GET XMIT SERV ADDR NEXT
011150  012710  000200          MOV     #200,(R0)
011154  012600                  MOV     (SP)+,R0                ;RESTORE
011156  000205                  RTS     R5


                        ;;*****************************************************************
                        ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
                        ;RETURN    IF TIMED OUT
                        ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
                        ;;*****************************************************************

011160  012537  011240  TIMER:  MOV     (R5)+,FLGHLD    ;GET FLAG
011164  012537  011242          MOV     (R5)+,CTHLD     ;GET COUNT THAT FLAG SHOULD GO TO

011170  012701  000010          MOV     #10,R1
011174  012700  177777  4$:     MOV     #-1,R0
011200  027737  000034  011242 1$:     CMP     @FLGHLD,CTHLD   ;RESPONDING?
011206  001411                  BEQ     2$              ;BR IF YES
011210  004737  010744          JSR     PC,EXAMKB
011214  005300                  DEC     R0              ;ELSE DEC COUNTER
011216  001370                  BNE     1$              ;BR IF NOT TIMED OUT
011220  005237  001200          INC     $DEVCT          ;FOR APT
011224  005301                  DEC     R1
011226  001362                  BNE     4$
011230  000402                  BR      .+6             ;ELSE TIMED OUT

011232  062705  000004  2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
011236  000205                  RTS     R5

011240  000000          FLGHLD: 0                       ;FLAG
011242  000000          CTHLD:  0                       ;COUNT THAT FLAG MUST REACH
```

```
2747                                    ;;*****************************************************************
2748                                    ;WATCHDOG TIMER TO PREVENT DX0 FROM ENTERING AN ENDLESS WAIT LOOP
2749                                    ;RETURN     IF TIMED OUT
2750                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
2751                                    ;;*****************************************************************
2752
2753  011244  005037  011336           TIMD0:  CLR     D0TMO
2754  011250  012737  000010  011334           MOV     #10,CTREG1
2755  011256  012737  177777  011332   4$:     MOV     #-1,CTREG
2756  011264  005737  003122           1$:     TST     FIN             ;FINISHED?
2757  011270  001015                           BNE     2$              ;BR IF YES
2758  011272  004737  010744                   JSR     PC,EXAMKB
2759  011276  005337  011332                   DEC     CTREG           ;ELSE DEC CTR
2760  011302  001370                           BNE     1$              ;BR IF NOT TIMED OUT
2761  011304  005237  001200                   INC     $DEVCT          ;FOR APT
2762  011310  005337  011334                   DEC     CTREG1
2763  011314  001360                           BNE     4$
2764  011316  005237  011336                   INC     D0TMO           ;SET FLAG
2765  011322  000402                           BR      .+6             ;TIMED OUT
2766
2767  011324  062716  000004           2$:     ADD     #4,(SP)         ;BUMP RET
2768  011330  000207                           RTS     PC
2769
2770  011332  000000                   CTREG:  0
2771  011334  000000                   CTREG1: 0
2772  011336  000000                   D0TMO:  0                       ;1 = TIMEOUT OCCURED
2773
2774
2775                                    ;;*****************************************************************
2776                                    ;WATCHDOG TIMER TO PREVENT DX1 FROM ENTERING AN ENDLESS WAIT LOOP
2777                                    ;RETURN     IF TIMED OUT
2778                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
2779                                    ;;*****************************************************************
2780
2781  011340  005037  011426           TIMD1:  CLR     D1TMO
2782  011344  012737  000010  011334           MOV     #10,CTREG1
2783  011352  012737  177777  011332   4$:     MOV     #-1,CTREG
2784  011360  005737  003122           1$:     TST     FIN             ;FINISHED?
2785  011364  001015                           BNE     2$              ;BR IF YES
2786  011366  004737  010744                   JSR     PC,EXAMKB
2787  011372  005337  011332                   DEC     CTREG           ;ELSE DEC CTR
2788  011376  001370                           BNE     1$              ;BR IF NOT TIMED OUT
2789  011400  005237  001200                   INC     $DEVCT          ;FOR APT
2790  011404  005337  011334                   DEC     CTREG1
2791  011410  001360                           BNE     4$
2792  011412  005237  011426                   INC     D1TMO           ;SET FLAG
2793  011416  000402                           BR      .+6             ;TIMED OUT
2794
2795  011420  062716  0000C4           2$:     ADD     #4,(SP)         ;BUMP RET
2796  011424  000207                           RTS     PC
2797
2798  011426  000000                   D1TMO:  0                       ;1 = TIMEOUT OCCURED
2799
```

```
                        ;;*****************************************************************
                        ;       GENERAL TIMER
                        ;;*****************************************************************

011430  012700  177777  TIMER1: MOV     #-1,R0
011434  005300                  DEC     R0
011436  001376                  BNE     .-2
011440  000207                  RTS     PC

                        ;;*****************************************************************
                        ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
                        ;RETURN     IF TIMED OUT
                        ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
                        ;;*****************************************************************

011442  012537  011522  TIMER2: MOV     (R5)+,REGHLD    ;GET REG
011446  012537  011524          MOV     (R5)+,BITHLD    ;GET BIT TO BE TESTED

011452  012701  000010          MOV     #10,R1
011456  012700  177777  4$:     MOV     #-1,R0
011462  033777  011524  000032 1$:     BIT     BITHLD,@REGHLD  ;BIT THERE?
011470  001011                  BNE     2$              ;BR IF YES
011472  004737  010744          JSR     PC,EXAMKB
011476  005300                  DEC     R0              ;ELSE DEC COUNTER
011500  001370                  BNE     1$              ;BR IF NOT TIMED OUT
011502  005237  001200          INC     $DEVCT          ;FOR APT
011506  005301                  DEC     R1
011510  001362                  BNE     4$
011512  000402                  BR      .+6             ;ELSE TIMED OUT

011514  062705  000004  2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
011520  000205                  RTS     R5

011522  000000          REGHLD: 0                       ;DEVICE REG
011524  000000          BITHLD: 0                       ;DEV REG BIT TO BE TESTED


                        ;;*****************************************************************
                        ;ROUTINE TO CLEAR BOTH BAD TRACK/SECTOR TABLES
                        ;;*****************************************************************

011526  012700  002624  CLRBAD: MOV     #D0BAD,R0
011532  005020          1$:     CLR     (R0)+
011534  020027  002650          CMP     R0,#D0BAD+20.
011540  001374                  BNE     1$

011542  012700  003074          MOV     #D1BAD,R0
011546  005020          2$:     CLR     (R0)+
011550  020027  003120          CMP     R0,#D1BAD+20.
011554  001374                  BNE     2$

011556  000207                  RTS     PC
```

```
2853                                    ;;*****************************************************************
2854                                    ;LOAD DX0 BAD TRACK/SECTOR TABLE
2855                                    ;;*****************************************************************
2856
2857  011560  010046                   LDOBAD: MOV     R0,-(SP)        ;SAVE
2858
2859  011562  012700  002624                   MOV     #DOBAD,R0
2860  011566  005710                   1$:     TST     (R0)            ;ENTRY PRESENT?
2861  011570  001003                           BNE     2$              ;BR IF YES
2862  011572  013710  177174                   MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
2863  011576  000405                           BR      3$              ;EXIT
2864
2865  011600  005720                   2$:     TST     (R0)+           ;BUMP PTR
2866  011602  020027  002650                   CMP     R0,#DOBAD+20.   ;AT END OF TABLE?
2867  011606  001367                           BNE     1$              ;BR IF NO
2868  011610  104104                           ERROR   104             ;10 BAD SECTORS...REPLACE
2869  011612  012600                   3$:     MOV     (SP)+,R0        ;RESTORE
2870  011614  000207                           RTS     PC
2871
2872
2873
2874
2875
2876                                    ;;*****************************************************************
2877                                    ;CHECK DX0 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
2878                                    ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
2879                                    ;;*****************************************************************
2880
2881  011616  010046                   CKOBAD: MOV     R0,-(SP)        ;SAVE
2882  011620  010146                           MOV     R1,-(SP)
2883
2884  011622  013701  002406                   MOV     DOTRK,R1
2885  011626  000301                           SWAB    R1
2886  011630  053701  002410                   BIS     DOSEC,R1
2887
2888  011634  012700  002624                   MOV     #DOBAD,R0
2889  011640  020027  002650           1$:     CMP     R0,#DOBAD+20.   ;END OF TABLE?
2890  011644  001405                           BEQ     2$              ;BR IF YES
2891  011646  005710                           TST     (R0)            ;ELSE ANY ENTRY?
2892  011650  001403                           BEQ     2$              ;BR IF NO
2893  011652  022001                           CMP     (R0)+,R1        ;ELSE COMPARE?
2894  011654  001371                           BNE     1$              ;BR IF NO
2895  011656  000402                           BR      3$              ;ELSE DONT BUMP RET ADDR
2896
2897  011660  062705  000002           2$:     ADD     #2,R5           ;BUMP RET ADDR
2898  011664  012601                   3$:     MOV     (SP)+,R1        ;RESTORE
2899  011666  012600                           MOV     (SP)+,R0
2900  011670  000205                           RTS     R5
```

```
                        ;;*****************************************************************
                        ;LOAD DX1 BAD TRACK/SECTOR TABLE
                        ;;*****************************************************************

011672  010046          LD1BAD: MOV     R0,-(SP)        ;SAVE

011674  012700  003074          MOV     #D1BAD,R0
011700  005710          1$:     TST     (R0)            ;ENTRY PRESENT?
011702  001003                  BNE     2$              ;BR IF YES
011704  013710  177174          MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
011710  000405                  BR      3$              ;EXIT

011712  005720          2$:     TST     (R0)+           ;BUMP PTR
011714  020027  003120          CMP     R0,#D1BAD+20.   ;AT END OF TABLE?
011720  001367                  BNE     1$              ;BR IF NO
011722  104105                  ERROR   105             ;10 BAD SECTORS...REPLACE
011724  012600          3$:     MOV     (SP)+,R0        ;RESTORE
011726  000207                  RTS     PC



                        ;;*****************************************************************
                        ;CHECK DX1 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
                        ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
                        ;;*****************************************************************

011730  010046          CK1BAD: MOV     R0,-(SP)        ;SAVE
011732  010146                  MOV     R1,-(SP)

011734  013701  002656          MOV     D1TRK,R1
011740  000301                  SWAB    R1
011742  053701  002660          BIS     D1SEC,R1

011746  012700  003074          MOV     #D1BAD,R0
011752  020027  003120  1$:     CMP     R0,#D1BAD+20.   ;END OF TABLE?
011756  001405                  BEQ     2$              ;BR IF YES
011760  005710                  TST     (R0)            ;ELSE ANY ENTRY?
011762  001403                  BEQ     2$              ;BR IF NO
011764  022001                  CMP     (R0)+,R1        ;ELSE COMPARE?
011766  001371                  BNE     1$              ;BR IF NO
011770  000402                  BR      3$              ;ELSE DONT BUMP RET ADDR

011772  062705  000002  2$:     ADD     #2,R5           ;BUMP RET ADDR
011776  012601          3$:     MOV     (SP)+,R1        ;RESTORE
012000  012600                  MOV     (SP)+,R0
012002  000205                  RTS     R5
```

```
                                        .SBTTL  INTERRUPT HANDLERS
                                        ;;*****************************************************************
                                        ; GENERAL SERVICE ROUTINE TO BUMP 'INTFLG'
                                        ; CALLING ROUTINE WILL KNOW WHAT TO DO
                                        ;;*****************************************************************

012004  005237  012012          INTSRV: INC     INTFLG
012010  000002                          RTI
012012  000000                  INTFLG: 0


                                        ;;*****************************************************************
                                        ;PRINTER  INTERRUPT HANDLER: VALUES FROM 40 THRU 176.
                                        ;;*****************************************************************

012014  013737  177514  012144  PRSRV:  MOV     PRS,SPRS                ;STORE
012022  005737  012144                  TST     SPRS            ;ERROR?
012026  100002                          BPL     1$              ;BR IF NO
012030  104007                          ERROR   7               ;PRINTER STATUS ERROR
012032  000002                          RTI

012034  013737  012140  177516  1$:     MOV     PRCHR,PRB       ;ELSE PRINT CHAR
012042  023727  012140  000015          CMP     PRCHR,#CR       ;JUST DID CR?
012050  001413                          BEQ     2$              ;BR IF YES
012052  023727  012140  000012          CMP     PRCHR,#LF       ;JUST DID LF?
012060  001413                          BEQ     3$              ;BR IF YES
012062  023727  012140  000176          CMP     PRCHR,#176      ;JUST DID LAST CHAR?
012070  001413                          BEQ     4$              ;BR IF YES
012072  005237  012140                  INC     PRCHR           ;ELSE BUMP CHAR FOR NEXT INTERRUPT
012076  000417                          BR      5$              ;EXIT

012100  012737  000012  012140  2$:     MOV     #LF,PRCHR       ;DO LF NEXT
012106  000413                          BR      5$
012110  012737  000040  012140  3$:     MOV     #40,PRCHR       ;DO SPACE NEXT
012116  000407                          BR      5$
012120  012737  000015  012140  4$:     MOV     #CR,PRCHR       ;DO CR NEXT & START OVER
012126  005237  012142                  INC     LINCT
012132  005237  001200                  INC     $DEVCT          ;FOR APT
012136  000002                  5$:     RTI

012140  000000                  PRCHR:  0                       ;CHAR TO BE PRINTED
012142  000000                  LINCT:  0                       ;LINE CTR
012144  000000                  SPRS:   0                       ;STORE PRINTER CSR
012146  000000                  PRPORT: 0                       ;SET IF DOING 5 LINES ONLY
```

```
                                        ;;*****************************************************************
                                        ;CLUSTER TERMINAL #1 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
                                        ;;*****************************************************************

012150  042737  000100  176504  TP1SRV: BIC     #IE,TP1S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
012156  013737  012244  176506          MOV     T1CHR,TP1B      ;PRINT CHAR
012164  000002                        RTI

012166  013737  176502  012246  TK1SRV: MOV     TK1B,T1HLD      ;STORE CHAR
012174  023737  012246  012244          CMP     T1HLD,T1CHR     ;OK?
012202  001401                        BEQ     1$              ;BR IF YES
012204  104010                        ERROR   10              ;CHAR COMP ERROR

012206  023727  012244  000377  1$:     CMP     T1CHR,#377      ;LAST CHAR?
012214  001403                        BEQ     2$              ;BR IF YES
012216  005237  012244                INC     T1CHR           ;ELSE BUMP
012222  000404                        BR      3$

012224  005037  012244          2$:     CLR     T1CHR           ;START OVER
012230  005237  001200                INC     $DEVCT          ;FOR APT
012234  052737  000100  176504  3$:     BIS     #IE,TP1S        ;ALLOW PRINTER INTERR
012242  000002                        RTI

012244  000000                T1CHR:  0                       ;CHAR TO XMITT
012246  000000                T1HLD:  0                       ;REC'D CHAR


                                        ;;*****************************************************************
                                        ;CLUSTER TERMINAL #2 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
                                        ;;*****************************************************************

012250  042737  000100  176514  TP2SRV: BIC     #IE,TP2S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
012256  013737  012344  176516          MOV     T2CHR,TP2B      ;PRINT CHAR
012264  000002                        RTI

012266  013737  176512  012346  TK2SRV: MOV     TK2B,T2HLD      ;STORE CHAR
012274  023737  012346  012344          CMP     T2HLD,T2CHR     ;OK?
012302  001401                        BEQ     1$              ;BR IF YES
012304  104011                        ERROR   11              ;CHAR COMP ERROR

012306  023727  012344  000377  1$:     CMP     T2CHR,#377      ;LAST CHAR?
012314  001403                        BEQ     2$              ;BR IF YES
012316  005237  012344                INC     T2CHR           ;ELSE BUMP
012322  000404                        BR      3$

012324  005037  012344          2$:     CLR     T2CHR           ;START OVER
012330  005237  001200                INC     $DEVCT          ;FOR APT
012334  052737  000100  176514  3$:     BIS     #IE,TP2S        ;ALLOW PRINTER INTERR
012342  000002                        RTI

012344  000000                T2CHR:  0                       ;CHAR TO XMITT
012346  000000                T2HLD:  0                       ;REC'D CHAR
```

```
3047
3048                                    ;;*****************************************************************
3049                                    ;CLUSTER TERMINAL #3 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3050                                    ;;*****************************************************************
3051
3052  012350  042737  000100  176524  TP3SRV: BIC     #IE,TP3S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3053  012356  013737  012444  176526          MOV     T3CHR,TP3B      ;PRINT CHAR
3054  012364  000002                         RTI
3055
3056  012366  013737  176522  012446  TK3SRV: MOV     TK3B,T3HLD      ;STORE CHAR
3057  012374  023737  012446  012444          CMP     T3HLD,T3CHR     ;OK?
3058  012402  001401                         BEQ     1$              ;BR IF YES
3059  012404  104012                         ERROR   12              ;CHAR COMP ERROR
3060
3061  012406  023727  012444  000377  1$:     CMP     T3CHR,#377      ;LAST CHAR?
3062  012414  001403                         BEQ     2$              ;BR IF YES
3063  012416  005237  012444                 INC     T3CHR           ;ELSE BUMP
3064  012422  000404                         BR      3$
3065
3066  012424  005037  012444          2$:     CLR     T3CHR           ;START OVER
3067  012430  005237  001200                 INC     $DEVCT          ;FOR APT
3068  012434  052737  000100  176524  3$:     BIS     #IE,TP3S        ;ALLOW PRINTER INTERR
3069  012442  000002                         RTI
3070
3071  012444  000000          T3CHR:  0               ;CHAR TO XMITT
3072  012446  000000          T3HLD:  0               ;REC'D CHAR
3073
3074
3075                                    ;;*****************************************************************
3076                                    ;ASYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3077                                    ;;*****************************************************************
3078
3079  012450  042737  000100  176614  AMXSRV: BIC     #IE,AMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3080  012456  013737  012544  176616          MOV     COMCHR,AMXB     ;XMITT CHAR
3081  012464  000002                         RTI
3082
3083  012466  013737  176612  012546  AMRSRV: MOV     AMRB,COMHLD     ;STORE CHAR
3084  012474  023737  012546  012544          CMP     COMHLD,COMCHR   ;OK?
3085  012502  001401                         BEQ     1$              ;BR IF YES
3086  012504  104013                         ERROR   13              ;CHAR COMPARE ERROR
3087
3088  012506  023727  012544  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
3089  012514  001403                         BEQ     2$              ;BR IF YES
3090  012516  005237  012544                 INC     COMCHR          ;ELSE BUMP
3091  012522  000404                         BR      3$
3092
3093  012524  005037  012544          2$:     CLR     COMCHR          ;START OVER
3094  012530  005237  001200                 INC     $DEVCT          ;FOR APT
3095  012534  052737  0001C0  176614  3$:     BIS     #IE,AMXC        ;ALLOW XMIT INTERR
3096  012542  000002                         RTI
3097
3098  012544  000000          COMCHR: 0               ;XMIT CHAR
3099  012546  000000          COMHLD: 0               ;REC'D CHAR
```

```
3100
3101                                    ;;*****************************************************************
3102                                    ;SYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 27 THRU 377.
3103                                    ;;*****************************************************************
3104
3105  012550  042737  000100  176624  SMXSRV: BIC     #IE,SMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3106  012556  013737  012544  176626          MOV     COMCHR,SMXB     ;XMITT CHAR
3107  012564  023727  012544  000377          CMP     COMCHR,#377     ;LAST CHAR?
3108  012572  001020                          BNE     2$              ;BR IF NO
3109
3110  012574  005737  012636                  TST     LSTCHR
3111  012600  001010                          BNE     1$
3112  012602  005237  012636                  INC     LSTCHR
3113  012606  052737  000100  176624          BIS     #IE,SMXC        ;ALLOW RDY INTR SO CAN SHUT DOWN FAST
3114  012614  005237  001200                  INC     $DEVCT          ;FOR APT
3115  012620  000002                          RTI
3116
3117  012622  042737  000020  176624  1$:     BIC     #SEND,SMXC      ;NO MORE TO SEND
3118  012630  005237  001200                  INC     $DEVCT          ;FOR APT
3119  012634  000002                  2$:     RTI
3120
3121  012636  000000                  LSTCHR: 0                       ;SET WHEN LAST CHAR XMIT
3122
3123
3124
3125
3126
3127  012640  013737  176622  012546  SMRSRV: MOV     SMRB,COMHLD     ;STORE CHAR
3128  012646  123727  012546  000026          CMPB    COMHLD,#026     ;IGNORE SYNC CHARS
3129  012654  001430                          BEQ     4$
3130  012656  023737  012546  012544          CMP     COMHLD,COMCHR   ;OK?
3131  012664  001401                          BEQ     1$              ;BR IF YES
3132  012666  104014                          ERROR   14              ;CHAR COMPARE ERROR
3133
3134  012670  023727  012544  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
3135  012676  001403                          BEQ     2$              ;BR IF YES
3136  012700  005237  012544                  INC     COMCHR          ;ELSE BUMP
3137  012704  000411                          BR      3$
3138
3139  012706  012737  177777  012544  2$:     MOV     #-1,COMCHR      ;INDICATE DONE
3140  012714  042737  000100  176620          BIC     #IE,SMRC        ;ALL DONE
3141  012722  005237  001200                  INC     $DEVCT          ;FOR APT
3142  012726  000403                          BR      4$
3143
3144  012730  052737  000100  176624  3$:     BIS     #IE,SMXC        ;ALLOW XMIT INTERR
3145  012736  000002                  4$:     RTI
3146
```

```
3147
3148                                    ;;*****************************************************************
3149                                    ;                   DISK INTERRUPT HANDLERS
3150                                    ;ALL DISK INTERRUPTS ENTER THRU RXSRV & DISPATCH TO THE
3151                                    ;CURRENT SERVICE ROUTINE POINTED TO BY:
3152                                    ;                   D0DISP FOR DX0 INTERRUPTS
3153                                    ;                   D1DISP FOR DX1 INTERRUPTS
3154                                    ;WILL GENERALLY BE IN THE ORDER OF SERVICE ROUTINES BELOW.
3155                                    ;;*****************************************************************
3156
3157  012740  105737  177170           RXSRV:  TSTB    RXCS            ;CONTR RDY?
3158  012744  100413                           BMI     1$              ;BR IF YES
3159  012746  013737  177170  013050           MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
3160  012754  013737  177172  013052           MOV     RXES,SRXES
3161  012762  013737  177174  013054           MOV     RXSA,SRXSA
3162  012770  104015                           ERROR   15              ;UNEXPECTED INTERRUPT
3163  012772  000002                           RTI                     ;GO BACK
3164
3165  012774  005737  177170           1$:     TST     RXCS            ;ERROR?
3166  013000  100011                           BPL     2$              ;BR IF NO
3167
3168  013002  013737  177170  013050           MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
3169  013010  013737  177172  013052           MOV     RXES,SRXES
3170  013016  013737  177174  013054           MOV     RXSA,SRXSA
3171
3172  013024  032737  000020  177170  2$:     BIT     #USEL,RXCS      ;CHECK UNIT SELECT BIT
3173  013032  001003                           BNE     D1INT           ;BR IF INTER FROM DX1
3174
3175  013034  000177  000000                   JMP     @D0DISP         ;ELSE DISPATCH TO DX0 SERVICE ROUTINE
3176  013040  000000           D0DISP: 0
3177
3178  013042  000177  000000   D1INT:  JMP     @D1DISP         ;DX1 INTERRUPT DISPATCH
3179  013046  000000           D1DISP: 0
3180
3181
3182  013050  000000           SRXCS:  0                       ;SAVE RXCS
3183  013052  000000           SRXES:  0                       ;SAVE RXES
3184  013054  000000           SRXSA:  0                       ;SAVE RXSA
3185
```

```
3186
3187                                    ;;*****************************************************************
3188                                    ; HANDLER TO FILL THE DRIVE BUFFER FOR DX0
3189                                    ;& POINT TO THE WRITE SECTOR HANDLER.
3190                                    ;;*****************************************************************
3191
3192  013056  012703  000100           DOFBUF: MOV     #100,R3         ;WORD CT
3193  013062  013701  002406                   MOV     DOTRK,R1
3194  013066  000301                           SWAB    R1
3195  013070  053701  002410                   BIS     DOSEC,R1
3196  013074  012737  013122  013040           MOV     #DOWSEC,DODISP  ;POINT TO WRITE SECTOR AFTER RTI
3197  013102  012737  000101  177170           MOV     #<FBUF!IE>,RXCS ;ISSUE FILL BUFFER CMD
3198  013110  010137  177172           1$:     MOV     R1,RXDB         ;FILL ENTIRE SECTOR WITH ITS ADDRESS
3199  013114  005303                           DEC     R3
3200  013116  001374                           BNE     1$
3201  013120  000002                           RTI
3202
3203
3204
3205
3206
3207                                    ;;*****************************************************************
3208                                    ;HANDLER TO WRITE THE SECTOR FOR DX0 & POINT TO WRITE CHECK.
3209                                    ;;*****************************************************************
3210
3211  013122  013701  002406           DOWSEC: MOV     DOTRK,R1
3212  013126  000301                           SWAB    R1
3213  013130  053701  002410                   BIS     DOSEC,R1
3214  013134  010137  177174                   MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
3215  013140  012737  013176  013040           MOV     #DOWCHK,DODISP  ;POINT TO WRITE CHECK HANDLER
3216  013146  023727  003120  000005           CMP     DXTST,#5
3217  013154  001004                           BNE     1$
3218  013156  012737  000115  177170           MOV     #<WDDSEC!IE>,RXCS       ;ONLY FOR WRITE DELETED DATA TEST
3219  013164  000403                           BR      2$
3220  013166  012737  000105  177170  1$:      MOV     #<WSEC!IE>,RXCS ;ISSUE WRITE SECTOR COMMAND
3221  013174  000002                  2$:      RTI
3222
```

```
3223                                    ;;*****************************************************************
3224                                    ;HANDLER TO WRITE CHECK ON DX0.
3225                                    ;WILL GO TO FILL BUFFER TO DO SAME SECTOR ON SOFT ERROR
3226                                    ;OR NEXT SECTOR ON NO ERROR OR HARD ERROR.
3227                                    ;WILL GO TO READ SECTOR AFTER ALL SECTORS ON TRACK ARE WRITTEN.
3228                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3229                                    ;;*****************************************************************
3230
3231  013176  005737  177170       DOWCHK: TST     RXCS            ;ERROR?
3232  013202  100024                        BPL     1$              ;BR IN NO
3233  013204  023727  002404  000012        CMP     DOERR,#10.      ;10 ERRORS?
3234  013212  001072                        BNE     6$              ;BR IF NO & TRY AGAIN
3235  013214  004737  011560                JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
3236  013220  032737  000020  013052        BIT     #RNF,SRXES      ;SEEK ERROR?
3237  013226  001410                        BEQ     7$              ;BR IF NO
3238  013230  104064                        ERROR   64              ;HARD SEEK ERROR WRITE SECTOR
3239  013232  012737  013310  013040        MOV     #2$,DODISP      ;GO TO 2$ AFTER RESTORE
3240  013240  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
3241  013246  000002                        RTI
3242
3243  013250  104016                7$:     ERROR   16              ;HARD ERROR WRITE SECTOR
3244  013252  000416                        BR      2$              ;GO TO NEXT SECTOR
3245
3246  013254  005737  002404        1$:     TST     DOERR           ;ANY PREV ERRORS?
3247  013260  001413                        BEQ     2$              ;BR IF NO
3248  013262  005237  026054                INC     TSERR           ;TOTAL SOFT ERR CT
3249  013266  005237  026060                INC     PSERR           ;PASS  SOFT ERR CT
3250  013272  032737  000020  013052        BIT     #RNF,SRXES      ;SEEK ERROR?
3251  013300  001402                        BEQ     8$              ;BR IF NO
3252  013302  104065                        ERROR   65              ;SOFT SEEK ERROR WRITE SECTOR
3253  013304  000401                        BR      2$
3254  013306  104017                8$:     ERROR   17              ;SOFT ERROR WRITE SECTOR
3255
3256  013310  005037  002404        2$:     CLR     DOERR
3257  013314  023727  002410  000031        CMP     DOSEC,#31       ;LAST ODD SECTOR?
3258  013322  001004                        BNE     3$              ;BR IF NO
3259  013324  012737  000002  002410        MOV     #2,DOSEC        ;ELSE DO EVEN SECTORS NOW
3260  013332  000407                        BR      4$
3261  013334  023727  002410  000032  3$:   CMP     DOSEC,#32       ;LAST EVEN SECTOR?
3262  013342  001405                        BEQ     5$              ;BR IF YES
3263  013344  062737  000002  002410        ADD     #2,DOSEC        ;ELSE BUMP SECTOR
3264  013352  000137  013056        4$:     JMP     DOFBUF          ;GO TO FILL BUFFER & DO ANOTHER
3265
3266  013356  012737  000001  002410  5$:   MOV     #1,DOSEC        ;ALL SECTORS DONE...CHECK DATA
3267  013364  005037  002404                CLR     DOERR
3268  013370  005037  002412                CLR     DOCMER
3269  013374  000137  013432                JMP     DORSEC          ;GO TO READ SECTOR HANDLER
3270
3271  013400  005237  0024C4        6$:     INC     DOERR
3272  013404  032737  000020  013052        BIT     #RNF,SRXES      ;SEEK ERROR?
3273  013412  001757                        BEQ     4$              ;BR IN NO
3274  013414  012737  013056  013040        MOV     #DOFBUF,DODISP  ;GOTO FILL BUFF AFTER RESTORE
3275  013422  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
3276  013430  000002                        RTI
```

```
3277
3278
3279                                    ;;*****************************************************************
3280                                    ;HANDLER TO READ A SECTOR ON DX0 & POINT TO CHECK.
3281                                    ;;*****************************************************************
3282
3283  013432  013701  002406    DORSEC: MOV     DOTRK,R1
3284  013436  000301                    SWAB    R1
3285  013440  053701  002410            BIS     DOSEC,R1
3286  013444  010137  177174            MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR
3287
3288  013450  012737  013466  013040    MOV     #DORCHK,DODISP  ;POINT TO READ CHECK HANDLER
3289  013456  012737  000107  177170    MOV     #<RSEC!IE>,RXCS ;ISSUE READ SECTOR COMMAND
3290  013464  000002                    RTI
3291
3292
3293                                    ;;*****************************************************************
3294                                    ;HANDLER TO READ CHECK ON DX0.
3295                                    ;WILL GO TO NEXT SECTOR/TRK IF HARD ERROR.
3296                                    ;WILL GO TO EMPTY BUFFER IF NO ERROR OR SOFT ERROR.
3297                                    ;WILL GO TO READ SECTOR (SAME) ON ERROR.
3298                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3299                                    ;;*****************************************************************
3300
3301  013466  005737  177170    DORCHK: TST     RXCS            ;ERROR?
3302  013472  100054                    BPL     1$              ;BR IF NO
3303  013474  023727  002404  000012    CMP     DOERR,#10.      ;10 ERRORS?
3304  013502  001415                    BEQ     3$              ;BR IF YES
3305  013504  005237  002404            INC     DOERR           ;ELSE DO AGAIN
3306  013510  032737  000020  013052    BIT     #RNF,SRXES      ;SEEK ERROR?
3307  013516  001745                    BEQ     DORSEC          ;BR IF NO
3308  013520  012737  013432  013040    MOV     #DORSEC,DODISP  ;ELSE DO RESTORE & RET TO RD SEC
3309  013526  012737  000117  177170    MOV     #<RESTOR!IE>,RXCS
3310  013534  000002                    RTI
3311
3312  013536  004737  011560    3$:     JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
3313  013542  032737  000020  013052    BIT     #RNF,SRXES      ;SEEK ERROR?
3314  013550  001410                    BEQ     7$              ;BR IF NO
3315  013552  104066                    ERROR   66              ;HARD SEEK ERROR READ SECTOR
3316
3317  013554  012737  014454  013040    MOV     #DONEXT,DODISP  ;DO RESTORE
3318  013562  012737  000117  177170    MOV     #<RESTOR!IE>,RXCS
3319  013570  000002                    RTI
3320
3321  013572  032737  000010  013052  7$:     BIT     #CRC,SRXES      ;CRC ERROR?
3322  013600  001404                    BEQ     10$             ;BR IF NO
3323  013602  005237  002422            INC     DOCRC           ;ELSE SET FLAG
3324  013606  104106                    ERROR   106             ;HARD CRC ERROR
3325  013610  000423                    BR      2$              ;& GO TO EMP BUFF TO CHK DATA
3326
3327  013612  005037  002422    10$:    CLR     DOCRC
3328  013616  104020                    ERROR   20              ;HARD READ ERROR
3329  013620  000137  014454            JMP     DONEXT
```

```
3330
3331  013624  005737  002404          1$:   TST     DOERR           ;ANY PREV ERRORS?
3332  013630  001413                        BEQ     2$              ;BR IF NO
3333  013632  005237  026054                INC     TSERR           ;TOTAL SOFT ERR CT
3334  013636  005237  026060                INC     PSERR           ;PASS  SOFT ERR CT
3335  013642  032737  000020  013052        BIT     #RNF,SRXES      ;SEEK ERROR?
3336  013650  001402                        BEQ     9$              ;BR IF NO
3337  013652  104067                        ERROR   67              ;SOFT SEEK ERROR  DURING READ
3338  013654  000401                        BR      2$
3339  013656  104021                9$:     ERROR   21              ;SOFT ERROR  READ SECTOR
3340
3341  013660  005037  002404          2$:   CLR     DOERR
3342  013664  023727  003120  000005        CMP     DXTST,#5        ;DOING WRITE DELETED DATA TESTS?
3343  013672  001005                        BNE     6$              ;BR IF NO
3344  013674  032737  000040  177172        BIT     #DD,RXES        ; 'DELETED DATA' SET?
3345  013702  001001                        BNE     6$              ;BR IF YES
3346  013704  104052                        ERROR   52              ;DD NOT SET
3347
3348  013706  000137  013712          6$:   JMP     DOEBUF          ;GO TO EMPTY BUFFER
3349
3350
3351
3352
3353
3354
3355                                  ;;*******************************************************************
3356                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX0
3357                                  ;& POINT TO THE CHECK DATA HANDLER.
3358                                  ;;*******************************************************************
3359
3360  013712  012703  000100          DOEBUF: MOV     #100,R3         ;WORD CT
3361  013716  012701  002424                  MOV     #DOBUF,R1       ;BUFFER ADDRESS
3362  013722  012737  013750  013040          MOV     #DOCDAT,DODISP  ;POINT TO CHK DATA AFT INTER.
3363  013730  012737  000103  177170          MOV     #<EBUF!IE>,RXCS ;ISSUE EMPTY BUFFER CMD
3364  013736  013721  177172          1$:     MOV     RXDB,(R1)+      ;GET WORD & STORE IT
3365  013742  005303                        DEC     R3
3366  013744  001374                        BNE     1$
3367  013746  000002                        RTI
3368
3369
```

```
                                ;;*****************************************************************
                                ; HANDLER TO CHECK DATA AFTER EMPTY BUFFER CMD ON DX0.
                                ;WILL GO TO NEXT SECTOR HANDLER WHEN NO ERR OR HARD ERR.
                                ;OR TO READ SECTOR (SAME) WHEN SOFT ERR.
                                ;THIS ROUTINE IS ALSO ENTERED FROM A CRC ERROR FROM A READ COMMAND.
                                ;;*****************************************************************

013750  005037  002412          DOCDAT: CLR     DOCMER
013754  012703  000100                  MOV     #100,R3         ;WORD CTR
013760  013701  002406                  MOV     DOTRK,R1
013764  000301                          SWAB    R1
013766  053701  002410                  BIS     DOSEC,R1        ;R1 NOW HAS EXPECTED DATA
013772  012702  002424                  MOV     #DOBUF,R2       ;GET BUFFER ADDR
013776  020112                  1$:     CMP     R1,(R2)         ;COMPARE OK?
014000  001407                          BEQ     2$              ;BR IF YES
014002  010137  002416                  MOV     R1,DOEXP        ;ELSE SAVE
014006  011237  002420                  MOV     (R2),DOREC
014012  005237  002412                  INC     DOCMER
014016  000405                          BR      3$

014020  005303                  2$:     DEC     R3
014022  001403                          BEQ     3$
014024  062702  000002                  ADD     #2,R2
014030  000762                          BR      1$

014032  005737  002412          3$:     TST     DOCMER          ;ANY COMP ERR?
014036  001421                          BEQ     4$              ;BR IF NO
014040  005737  002422                  TST     DOCRC           ;HERE FROM CRC ERROR?
014044  001402                          BEQ     14$             ;BR IF NO
014046  104107                          ERROR   107             ;DATA CRC ERROR
014050  000431                          BR      5$

014052  023727  002414  000012  14$:    CMP     DOCERR,#10.     ;ELSE, 10 ERRS YET?
014060  001033                          BNE     7$              ;BR IF NO
014062  023727  003120  000003          CMP     DXTST,#3        ;DOING INIT TEST?
014070  001002                          BNE     8$              ;BR IN NO
014072  104053                          ERROR   53              ;INITIALIZE ERROR
014074  000417                          BR      5$

014076  104022                  8$:     ERROR   22              ;HARD ERROR  DATA COMPARE
014100  000415                          BR      5$

014102  005737  002422          4$:     TST     DOCRC           ;HERE FROM CRC ERROR?
014106  001402                          BEQ     15$             ;BR IF NO
014110  104110                          ERROR   110             ;CRC ERR WITH DATA OK
014112  000410                          BR      5$

014114  005737  002414          15$:    TST     DOCERR          ;ANY PREV ERR?
014120  001405                          BEQ     5$              ;BR IF NO
014122  005237  026054                  INC     TSERR           ;TOTAL SOFT ERR CT
014126  005237  026060                  INC     PSERR           ;PASS  SOFT ERR CT
014132  104023                          ERROR   23              ;SOFT ERROR  DATA COMP
```

```
3422
3423  014134  005037  002414          5$:     CLR     DOCERR
3424  014140  005037  002422                  CLR     DOCRC
3425  014144  000137  014454                  JMP     DONEXT
3426
3427  014150  005237  002414          7$:     INC     DOCERR
3428  014154  023727  003120  000003          CMP     DXTST,#3        ;DOING INIT TEST?
3429  014162  001402                          BEQ     12$             ;BR IF YES
3430  014164  000137  013432                  JMP     DORSEC          ;ELSE READ SECTOR OVER AGAIN
3431
3432  014170  000137  014612          12$:    JMP     DODUN           ;ALL DONE
3433
3434
3435
3436                                  ;;*****************************************************************
3437                                  ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX0.
3438                                  ;WILL GO TO READ SECTOR (NEXT) IF ALL SECTORS ON TRACK NOT DONE.
3439                                  ;WILL GO TO FILL BUFFER (NEXT TRACK) IF ALL SECTORS ON TRACK DONE.
3440                                  ;WILL SHUT OFF DRV INTERRUPTS IF ALL TRACKS DONE, WITH A MESSAGE.
3441                                  ;;*****************************************************************
3442
3443  014174  005037  002404          DONXT1: CLR     DOERR
3444  014200  005037  002412                  CLR     DOCMER
3445  014204  023727  002410  000031          CMP     DOSEC,#31       ;LAST ODD SECTOR?
3446  014212  001004                          BNE     1$              ;BR IF NO
3447  014214  012737  000002  002410          MOV     #2,DOSEC        ;ELSE DO EVEN SECTORS NOW
3448  014222  000407                          BR      2$
3449  014224  023727  002410  000032  1$:     CMP     DOSEC,#32       ;LAST EVEN SECTOR?
3450  014232  001405                          BEQ     3$              ;BR IF YES
3451  014234  062737  000002  002410          ADD     #2,DOSEC        ;ELSE BUMP SECTOR
3452  014242  000137  013432          2$:     JMP     DORSEC          ;GO READ SECTOR
3453
3454  014246  032777  010000  164664  3$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3455  014254  001411                          BEQ     6$
3456  014256  104401  017076                  TYPE    ,DRV0
3457  014262  104401  017551                  TYPE    ,TRK
3458  014266  013746  002406                  MOV     DOTRK,-(SP)     ;;SAVE DOTRK FOR TYPEOUT
3459  014272  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3460  014274  104401  017622                  TYPE    ,DUN
3461  014300  005237  003122          6$:     INC     FIN             ;SETUP TO GO TO DX1
3462  014304  023737  002406  003124          CMP     DOTRK,MAXTRK    ;LAST TRACK?
3463  014312  001406                          BEQ     4$              ;BR IF YES
3464  014314  005237  002406                  INC     DOTRK           ;BUMP TRACK
3465  014320  012737  000001  002410          MOV     #1,DOSEC        ;INIT SECTOR
3466  014326  000414                          BR      5$
3467
3468  014330  032777  010000  164602  4$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3469  014336  001406                          BEQ     7$
3470  014340  104401  017076                  TYPE    ,DRV0
3471  014344  104401  017556                  TYPE    ,PATT
3472  014350  104401  017622                  TYPE    ,DUN
3473  014354  005237  002400          7$:     INC     DOPAT           ;PATT DONE FLAG
3474  014360  042737  000100  177170  5$:     BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
3475  014366  000002                          RTI
```

```
3476
3477                                    ;;*****************************************************************
3478                                    ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX0.
3479                                    ;;*****************************************************************
3480
3481  014370  005037  002404           DONXT2: CLR     DOERR
3482  014374  005037  002412                   CLR     DOCMER
3483  014400  005237  003122                   INC     FIN             ;SETUP TO GO TO DX1
3484  014404  005237  002402                   INC     DOCNT
3485  014410  023737  002402  003126           CMP     DOCNT,MAXSK     ;DID ALL SEEKS?
3486  014416  001012                           BNE     1$              ;BR IF NO
3487  014420  032777  010000  164512           BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3488  014426  001406                           BEQ     1$
3489  014430  104401  017076                   TYPE    ,DRV0
3490  014434  104401  017571                   TYPE    ,RANDOM
3491  014440  104401  017622                   TYPE    ,DUN
3492  014444  042737  000100  177170  1$:     BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
3493  014452  000002                           RTI
3494
3495
3496
3497
3498
3499  014454  023727  003120  000001  DONEXT: CMP     DXTST,#1
3500  014462  001644                           BEQ     DONXT1
3501  014464  023727  003120  000002           CMP     DXTST,#2
3502  014472  001736                           BEQ     DONXT2
3503  014474  000446                           BR      DODUN
3504
```

```
                                        ;;*****************************************************************
                                        ;HANDLER TO SETUP INITIALIZE COMMAND TEST
                                        ;;*****************************************************************

014476  012737  046032  177174  DOINIT: MOV     #46032,RXSA     ;TRY TO FAKE DRIVE TO GO TO TRK 76 SEC 26
                                                                ;INIT SHOULD GO TO TRK 1, SEC 1.
014504  012737  014612  013040          MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
014512  012737  000111  177170          MOV     #<INITAL!IE>,RXCS       ;DO INIT COMMAND
014520  000002                          RTI

                                        ;;*****************************************************************
                                        ;HANDLERS TO SETUP RESTORE COMMAND TEST
                                        ;;*****************************************************************

014522  012737  014612  013040  DOREST: MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
014530  012737  000117  177170          MOV     #<RESTOR!IE>,RXCS
014536  000002                          RTI

014540  012737  014626  013046  D1REST: MOV     #D1DUN,D1DISP
014546  012737  000137  177170          MOV     #<RESTOR!IE!DX1>,RXCS
014554  000002                          RTI

                                        ;;*****************************************************************
                                        ;HANDLER TO SETUP READ STATUS COMMAND TEST
                                        ;;*****************************************************************

014556  012737  014612  013040  DOSTAT: MOV     #DODUN,DODISP   ;POINT TO DONE
014564  012737  000113  177170          MOV     #<RSTAT!IE>,RXCS        ;DO READ STATUS COMMAND
014572  000002                          RTI

                                        ;;*****************************************************************
                                        ;HANDLER TO START INVALID ADDRESS TESTS
                                        ;;*****************************************************************

014574  012737  014612  013040  DOINV:  MOV     #DODUN,DODISP   ;POINT TO DONE
014602  012737  000105  177170          MOV     #<WSEC!IE>,RXCS ;DO WRITE SECTOR COMMAND
014610  000002                          RTI

                                        ;;*****************************************************************
                                        ;HANDLERS TO FINISH INITIALIZE, RESTORE, WRITE DELETED DATA,
                                        ;READ STATUS & INVALID ADDRESS TESTS.
                                        ;;*****************************************************************

014612  005237  003122          DODUN:  INC     FIN             ;SET DONE FLAG
014616  042737  000100  177170          BIC     #IE,RXCS        ;DISABLE FURTHUR DX0 INTERRUPTS
014624  000002                          RTI

014626  005237  003122          D1DUN:  INC     FIN
014632  042737  000120  177170          BIC     #<IE!DX1>,RXCS
014640  000002                          RTI
```

```
3559
3560
3561                                    ;;*****************************************************************
3562                                    ; HANDLER TO FILL THE DRIVE BUFFER FOR DX1
3563                                    ;& POINT TO THE WRITE SECTOR HANDLER.
3564                                    ;;*****************************************************************
3565
3566  014642  012703  000100           D1FBUF: MOV     #100,R3         ;WORD CT
3567  014646  013701  002656                   MOV     D1TRK,R1
3568  014652  000301                           SWAB    R1
3569  014654  053701  002660                   BIS     D1SEC,R1
3570
3571  014660  005737  003132                   TST     COMP1           ;DONT SET BIT15 IN COMPATABILTIY TESTS
3572  014664  001005                           BNE     2$
3573  014666  005737  003134                   TST     COMP2
3574  014672  001002                           BNE     2$
3575  014674  052701  100000                   BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA
3576
3577  014700  012737  014726  013046   2$:     MOV     #D1WSEC,D1DISP  ;POINT TO WRITE SECTOR AFTER RTI
3578  014706  012737  000121  177170           MOV     #<FBUF!IE!DX1>,RXCS     ;ISSUE FILL BUFFER CMD
3579  014714  010137  177172           1$:     MOV     R1,RXDB         ;FILL ENTIRE SECTOR WITH ITS ADDRESS
3580  014720  005303                           DEC     R3              ;& BIT 15 SET
3581  014722  001374                           BNE     1$
3582  014724  000002                           RTI
3583
3584
3585                                    ;;*****************************************************************
3586                                    ;HANDLER TO WRITE THE SECTOR FOR DX1 & POINT TO WRITE CHECK.
3587                                    ;;*****************************************************************
3588
3589  014726  013701  002656           D1WSEC: MOV     D1TRK,R1
3590  014732  000301                           SWAB    R1
3591  014734  053701  002660                   BIS     D1SEC,R1
3592  014740  010137  177174                   MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
3593  014744  012737  014762  013046           MOV     #D1WCHK,D1DISP  ;POINT TO WRITE CHECK HANDLER
3594  014752  012737  000125  177170           MOV     #<WSEC!IE!DX1>,RXCS     ;ISSUE WRITE SECTOR COMMAND
3595  014760  000002                           RTI
3596
```

```
3597                                    ;;*****************************************************************
3598                                    ;HANDLER TO WRITE CHECK ON DX1.
3599                                    ;WILL GO TO FILL BUFFER TO DO SAME SECTOR ON SOFT ERROR
3600                                    ;OR NEXT SECTOR ON NO ERROR OR HARD ERROR.
3601                                    ;WILL GO TO READ SECTOR AFTER ALL SECTORS ON TRACK ARE WRITTEN.
3602                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3603                                    ;;*****************************************************************
3604
3605  014762  005737  177170       D1WCHK: TST     RXCS            ;ERROR?
3606  014766  100024                       BPL     1$              ;BR IN NO
3607  014770  023727  002654  000012       CMP     D1ERR,#10.      ;10 ERRORS?
3608  014776  001072                       BNE     6$              ;BR IF NO & TRY AGAIN
3609  015000  004737  011672               JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
3610  015004  032737  000020  013052       BIT     #RNF,SRXES      ;SEEK ERROR?
3611  015012  001410                       BEQ     7$              ;BR IF NO
3612  015014  104070                       ERROR   70              ;HARD SEEK ERROR WRITE SECTOR
3613  015016  012737  015074  013046       MOV     #2$,D1DISP      ;RET TO 2$ AFT RESTORE
3614  015024  012737  000137  177170       MOV     #<RESTOR!IE!DX1>,RXCS
3615  015032  000002                       RTI
3616
3617  015034  104024               7$:     ERROR   24              ;HARD ERROR WRITE SECTOR
3618  015036  000416                       BR      2$              ;GO TO NEXT SECTOR
3619
3620  015040  005737  002654       1$:     TST     D1ERR           ;ANY PREV ERRORS?
3621  015044  001413                       BEQ     2$              ;BR IF NO
3622  015046  005237  026054               INC     TSERR           ;TOTAL SOFT ERR CT
3623  015052  005237  026060               INC     PSERR           ;PASS  SOFT ERR CT
3624  015056  032737  000020  013052       BIT     #RNF,SRXES      ;SEEK ERROR?
3625  015064  001402                       BEQ     8$              ;BR IF NO
3626  015066  104071                       ERROR   71              ;SOFT SEEK ERROR WRITE SECTOR
3627  015070  000401                       BR      2$
3628  015072  104025               8$:     ERROR   25              ;SOFT ERROR WRITE SECTOR
3629
3630  015074  005037  002654       2$:     CLR     D1ERR
3631  015100  023727  002660  000031       CMP     D1SEC,#31       ;LAST ODD SECTOR?
3632  015106  001004                       BNE     3$              ;BR IF NO
3633  015110  012737  000002  002660       MOV     #2,D1SEC        ;ELSE DO EVEN SECTORS NOW
3634  015116  000407                       BR      4$
3635  015120  023727  002660  000032  3$:  CMP     D1SEC,#32       ;LAST EVEN SECTOR?
3636  015126  001405                       BEQ     5$              ;BR IF YES
3637  015130  062737  000002  002660       ADD     #2,D1SEC        ;ELSE BUMP SECTOR
3638  015136  000137  014642       4$:     JMP     D1FBUF          ;GO TO FILL BUFFER & DO ANOTHER
3639
3640  015142  012737  000001  002660  5$:  MOV     #1,D1SEC        ;ALL SECTORS DONE...CHECK DATA
3641  015150  005037  002654               CLR     D1ERR
3642  015154  005037  002662               CLR     D1CMER
3643  015160  000137  015216               JMP     D1RSEC          ;GO TO READ SECTOR HANDLER
3644
3645  015164  005237  002654       6$:     INC     D1ERR
3646  015170  032737  000020  013052       BIT     #RNF,SRXES      ;SEEK ERROR?
3647  015176  001757                       BEQ     4$              ;BR IF NO
3648  015200  012737  014642  013046       MOV     #D1FBUF,D1DISP  ;RET TO FILL BUFF AFT RESTORE
3649  015206  012737  000137  177170       MOV     #<RESTOR!IE!DX1>,RXCS
3650  015214  000002                       RTI
```

```
                        ;;*****************************************************************
                        ;HANDLER TO READ A SECTOR ON DX1 & POINT TO CHECK.
                        ;;*****************************************************************

015216 013701 002656    D1RSEC: MOV     D1TRK,R1
015222 000301                   SWAB    R1
015224 053701 002660            BIS     D1SEC,R1
015230 010137 177174            MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR

015234 012737 015252 013046     MOV     #D1RCHK,D1DISP  ;POINT TO READ CHECK HANDLER
015242 012737 000127 177170     MOV     #<RSEC!IE!DX1>,RXCS     ;ISSUE READ SECTOR COMMAND
015250 000002                   RTI


                        ;;*****************************************************************
                        ;HANDLER TO READ CHECK ON DX1.
                        ;WILL GO TO NEXT SECTOR/TRK IF HARD ERROR.
                        ;WILL GO TO EMPTY BUFFER IF NO ERROR OR SOFT ERROR.
                        ;WILL GO TO READ SECTOR (SAME) ON ERROR.
                        ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
                        ;;*****************************************************************

015252 005737 177170    D1RCHK: TST     RXCS            ;ERROR?
015256 100054                   BPL     1$              ;BR IF NO
015260 023727 002654 000012     CMP     D1ERR,#10.      ;10 ERRORS?
015266 001415                   BEQ     3$              ;BR IF YES
015270 005237 002654            INC     D1ERR           ;ELSE TRY AGAIN
015274 032737 000020 013052     BIT     #RNF,SRXES      ;SEEK ERROR?
015302 001745                   BEQ     D1RSEC          ;BR IF NO
015304 012737 015216 013046     MOV     #D1RSEC,D1DISP  ;ELSE DO RESTORE & RET TO RD SEC
015312 012737 000137 177170     MOV     #<RESTOR!IE!DX1>,RXCS
015320 000002                   RTI

015322 004737 011672    3$:     JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
015326 032737 000020 013052     BIT     #RNF,SRXES      ;SEEK ERROR?
015334 001410                   BEQ     7$              ;BR IF NO
015336 104072                   ERROR   72              ;HARD SEEK ERROR  DURING READ

015340 012737 016206 013046     MOV     #D1NEXT,D1DISP  ;DO RESTORE
015346 012737 000137 177170     MOV     #<RESTOR!IE!DX1>,RXCS
015354 000002                   RTI

015356 032737 000010 013052 7$: BIT     #CRC,SRXES      ;CRC ERROR?
015364 001404                   BEQ     10$             ;BR IF NO
015366 005237 002672            INC     D1CRC           ;ELSE SET FLAG
015372 104111                   ERROR   111             ;HARD CRC ERROR
015374 000423                   BR      2$              ;& GO TO EMP BUFF TO CHK DATA

015376 005037 002672    10$:    CLR     D1CRC
015402 104026                   ERROR   26              ;HARD READ ERROR
015404 000137 016206            JMP     D1NEXT
```

```
3704
3705  015410  005737  002654          1$:     TST     D1ERR           ;ANY PREV ERRORS?
3706  015414  001413                          BEQ     2$              ;BR IF NO
3707  015416  005237  026054                  INC     TSERR           ;TOTAL SOFT ERR CT
3708  015422  005237  026060                  INC     PSERR           ;PASS  SOFT ERR CT
3709  015426  032737  000020  013052          BIT     #RNF,SRXES      ;SEEK ERROR?
3710  015434  001402                          BEQ     9$              ;BR IF NO
3711  015436  104073                          ERROR   73              ;SOFT SEEK ERROR  DURING READ
3712  015440  000401                          BR      2$
3713  015442  104027                  9$:     ERROR   27              ;SOFT ERROR  READ SECTOR
3714
3715  015444  005037  002654          2$:     CLR     D1ERR
3716  015450  000137  015454                  JMP     D1EBUF          ;GO TO EMPTY BUFFER
3717
3718
3719
3720
3721
3722
3723                                  ;;*****************************************************************
3724                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX1
3725                                  ;& POINT TO THE CHECK DATA HANDLER.
3726                                  ;;*****************************************************************
3727
3728  015454  012703  000100          D1EBUF: MOV     #100,R3         ;WORD CT
3729  015460  012701  002674                  MOV     #D1BUF,R1       ;BUFFER ADDRESS
3730  015464  012737  015512  013046          MOV     #D1CDAT,D1DISP  ;POINT TO CHK DATA AFT INTER.
3731  015472  012737  000123  177170          MOV     #<EBUF!IE!DX1>,RXCS     ;ISSUE EMPTY BUFFER CMD
3732  015500  013721  177172          1$:     MOV     RXDB,(R1)+      ;GET WORD & STORE IT
3733  015504  005303                          DEC     R3
3734  015506  001374                          BNE     1$
3735  015510  000002                          RTI
3736
```

```
3737                                    ;;*****************************************************************
3738                                    ; HANDLER TO CHECK DATA AFTER EMPTY BUFFER CMD ON DX1.
3739                                    ;WILL GO TO NEXT SECTOR HANDLER WHEN NO ERR OR HARD ERR.
3740                                    ;OR TO READ SECTOR (SAME) WHEN SOFT ERR.
3741                                    ;THIS ROUTINE IS ALSO ENTERED FROM A CRC ERROR FROM A READ COMMAND.
3742                                    ;;*****************************************************************
3743
3744  015512  005037  002662           D1CDAT: CLR     D1CMER
3745  015516  012703  000100                   MOV     #100,R3         ;WORD CTR
3746  015522  013701  002656                   MOV     D1TRK,R1
3747  015526  000301                           SWAB    R1
3748  015530  053701  002660                   BIS     D1SEC,R1
3749
3750  015534  005737  003132                   TST     COMP1           ;DONT SET BIT15 IN COMPATABILITY TESTS
3751  015540  001005                           BNE     6$
3752  015542  005737  003134                   TST     COMP2
3753  015546  001002                           BNE     6$
3754  015550  052701  100000                   BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA
3755
3756  015554  012702  002674           6$:     MOV     #D1BUF,R2       ;GET BUFFER ADDR
3757  015560  020112                   1$:     CMP     R1,(R2)         ;COMPARE OK?
3758  015562  001407                           BEQ     2$              ;BR IF YES
3759  015564  010137  002666                   MOV     R1,D1EXP        ;ELSE SAVE
3760  015570  011237  002670                   MOV     (R2),D1REC
3761  015574  005237  002662                   INC     D1CMER
3762  015600  000405                           BR      3$
3763  015602  005303                   2$:     DEC     R3
3764  015604  001403                           BEQ     3$
3765  015606  062702  000002                   ADD     #2,R2
3766  015612  000762                           BR      1$
3767
3768  015614  005737  002662           3$:     TST     D1CMER          ;ANY COMP ERR?
3769  015620  001413                           BEQ     4$              ;BR IF NO
3770  015622  005737  002672                   TST     D1CRC           ;HERE FROM CRC ERROR?
3771  015626  001402                           BEQ     14$             ;BR IF NO
3772  015630  104112                           ERROR   112             ;DATA CRC ERROR
3773  015632  000423                           BR      5$
3774  015634  023727  002664  000012   14$:    CMP     D1CERR,#10.     ;ELSE, 10 ERRS YET?
3775  015642  001025                           BNE     7$              ;BR IF NO
3776  015644  104030                           ERROR   30              ;HARD ERROR  DATA COMP
3777  015646  000415                           BR      5$
3778
3779  015650  005737  002672           4$:     TST     D1CRC           ;HERE FROM CRC ERROR?
3780  015654  001402                           BEQ     15$             ;BR IF NO
3781  015656  104113                           ERROR   113             ;CRC ERR WITH DATA OK
3782  015660  000410                           BR      5$
3783  015662  005737  002664           15$:    TST     D1CERR          ;ANY ERR?
3784  015666  001405                           BEQ     5$
3785  015670  005237  026054                   INC     TSERR           ;TOTAL SOFT ERR CT
3786  015674  005237  026060                   INC     PSERR           ;PASS  SOFT ERR CT
3787  015700  104031                           ERROR   31              ;SOFT ERROR  DATA COMP
```

```
3788
3789  015702  005037  002664          5$:     CLR     D1CERR
3790  015706  005037  002672                  CLR     D1CRC
3791  015712  000137  016206                  JMP     D1NEXT
3792
3793  015716  005237  002664          7$:     INC     D1CERR
3794  015722  000137  015216                  JMP     D1RSEC          ;READ SECTOR OVER AGAIN
3795
3796
3797                                  ;;*****************************************************************
3798                                  ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX1.
3799                                  ;WILL GO TO READ SECTOR (NEXT) IF ALL SECTORS ON TRACK NOT DONE.
3800                                  ;WILL GO TO FILL BUFFER (NEXT TRACK) IF ALL SECTORS ON TRACK DONE.
3801                                  ;WILL SHUT OFF DRV INTERRUPTS IF ALL TRACKS DONE, WITH A MESSAGE.
3802                                  ;;*****************************************************************
3803
3804  015726  005037  002654          D1NXT1: CLR     D1ERR
3805  015732  005037  002662                  CLR     D1CMER
3806  015736  023727  002660  000031          CMP     D1SEC,#31       ;LAST ODD SECTOR?
3807  015744  001004                          BNE     1$              ;BR IF NO
3808  015746  012737  000002  002660          MOV     #2,D1SEC        ;ELSE DO EVEN SECTORS NOW
3809  015754  000407                          BR      2$
3810  015756  023727  002660  000032  1$:     CMP     D1SEC,#32       ;LAST EVEN SECTOR?
3811  015764  001405                          BEQ     3$              ;BR IF YES
3812  015766  062737  000002  002660          ADD     #2,D1SEC        ;ELSE BUMP SECTOR
3813  015774  000137  015216          2$:     JMP     D1RSEC          ;GO READ SECTOR
3814
3815  016000  032777  010000  163132  3$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3816  016006  001411                          BEQ     6$
3817  016010  104401  017103                  TYPE    ,DRV1
3818  016014  104401  017551                  TYPE    ,TRK
3819  016020  013746  002656                  MOV     D1TRK,-(SP)     ;;SAVE D1TRK FOR TYPEOUT
3820  016024  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3821  016026  104401  017622                  TYPE    ,DUN
3822  016032  005237  003122          6$:     INC     FIN             ;SETUP TO GO TO DX0
3823  016036  023737  002656  003124          CMP     D1TRK,MAXTRK    ;LAST TRACK?
3824  016044  001406                          BEQ     4$              ;BR IF YES
3825  016046  005237  002656                  INC     D1TRK           ;BUMP TRACK
3826  016052  012737  000001  002660          MOV     #1,D1SEC        ;INIT SECTOR
3827  016060  000414                          BR      5$
3828
3829  016062  032777  010000  163050  4$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3830  016070  001406                          BEQ     7$
3831  016072  104401  017103                  TYPE    ,DRV1
3832  016076  104401  017556                  TYPE    ,PATT
3833  016102  104401  017622                  TYPE    ,DUN
3834  016106  005237  002650          7$:     INC     D1PAT           ;PATT DONE FLAG
3835  016112  042737  000120  177170  5$:     BIC     #<IE!DX1>,RXCS  ;DISABLE DX1 INTERRUPTS
3836  016120  000002                          RTI
3837
```

```
3838
3839                                    ;;*****************************************************************
3840                                    ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX1.
3841                                    ;;*****************************************************************
3842
3843  016122  005037  002654           D1NXT2: CLR     D1ERR
3844  016126  005037  002662                   CLR     D1CMER
3845  016132  005237  003122                   INC     FIN             ;SETUP TO GO TO DX0
3846  016136  005237  002652                   INC     D1CNT
3847  016142  023737  002652  003126           CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
3848  016150  001012                           BNE     1$              ;BR IF NO
3849  016152  032777  010000  162760           BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3850  016160  001406                           BEQ     1$
3851  016162  104401  017103                   TYPE    ,DRV1
3852  016166  104401  017571                   TYPE    ,RANDOM
3853  016172  104401  017622                   TYPE    ,DUN
3854  016176  042737  000120  177170  1$:      BIC     #<IE!DX1>,RXCS  ;DISABLE DX1 INTERRUPTS
3855  016204  000002                           RTI
3856
3857
3858
3859
3860  016206  023727  003120  000001  D1NEXT: CMP     DXTST,#1
3861  016214  001644                           BEQ     D1NXT1
3862  016216  000741                           BR      D1NXT2
3863
```

```
3864
3865                            ;;*****************************************************************
3866                            ;THE FOLLOWING CODE, UP TO THE PROGRAM MESSAGES, IS A UTILITY TO COPY
3867                            ;THE DISKETTE CONTAINING ONLY THE BOOT & THE EXERCISER FROM DX0 TO DX1.
3868                            ;THE OPERATOR CAN TYPE 'P' TO DO ANOTHER COPY OR
3869                            ;TYPE '240G' TO ENTER NORMAL TESTING.
3870                            ;ENTIRE CODE, INCLUDING TEXT & ERROR TABLE, IS 400(10) WORDS.
3871                            ;;*****************************************************************
3872
3873  016220  005037  003130        COPY:   CLR     COPFLG                  ;DONT NEED FLAG ANY MORE
3874  016224  104401  020021                TYPE    ,COPMSG                 ;INSERT SCRATCH IN DX1 & TYPE P
3875  016230  000000                HALT                                    ;WAIT FOR PROCEED
3876
3877  016232  004737  016540                JSR     PC,GETSEC               ;CALCULATE MAX SECTORS TO BE COPIED
3878
3879  016236  005037  016532                CLR     ERFLG                   ;INITIALIZE
3880  016242  012737  000001  016534        MOV     #1,TRACK
3881  016250  012737  000001  016536        MOV     #1,SECT
3882  016256  105737  177170                TSTB    RXCS                    ;DONE?  (CTL RDY)
3883  016262  100375                BPL     .-4                             ;BR IF NO & TRY AGAIN
3884
3885  016264  004537  016610        1$:     JSR     R5,RDWR         ;READ DX0
3886  016270  000007                    RSEC
3887
3888  016272  005737  177170                TST     RXCS                    ;ERROR?
3889  016276  100011                BPL     2$                              ;BR IF NO
3890  016300  005237  016532                INC     ERFLG
3891  016304  023727  016532  000012        CMP     ERFLG,#10.              ;10 ERRORS?
3892  016312  001364                BNE     1$                              ;BR IF NO & TRY AGAIN
3893  016314  104100                ERROR   100                             ;HARD ERR READ SECTOR DX0
3894  016316  000000                HALT
3895  016320  000776                BR      .-2                             ;FORCE A RESTART FROM 250
3896
3897  016322  005037  016532        2$:     CLR     ERFLG
3898  016326  004537  016650                JSR     R5,EBUFF        ;EMPTY BUFFER INTO DOBUF
3899  016332  002424                    DOBUF
3900
3901  016334  004537  016610        3$:     JSR     R5,RDWR         ;WRITE DX1
3902  016340  000025                    <WSEC!DX1>
3903
3904  016342  005737  177170                TST     RXCS                    ;ERROR?
3905  016346  100011                BPL     4$                              ;BR IF NO
3906  016350  005237  016532                INC     ERFLG
3907  016354  023727  016532  000012        CMP     ERFLG,#10.              ;10 ERRORS?
3908  016362  001364                BNE     3$                              ;BR IF NO & TRY AGAIN
3909  016364  104101                ERROR   101                             ;HARD ERR WRITE SECTOR DX1
3910  016366  000000                HALT
3911  016370  000776                BR      .-2                             ;FORCE RESTART AT 250
```

```
3912
3913  016372  005037  016532          4$:   CLR     ERFLG
3914  016376  004537  016610                JSR     R5,RDWR         ;READ DX1
3915  016402  000027                           <RSEC!DX1>
3916
3917  016404  005737  177170                TST     RXCS                    ;ERROR?
3918  016410  100011                        BPL     5$                      ;BR IF NO
3919  016412  005237  016532                INC     ERFLG
3920  016416  023727  016532  000012        CMP     ERFLG,#10.              ;10 ERRORS?
3921  016424  001362                        BNE     4$                      ;BR IF NO & TRY AGAIN
3922  016426  104103                        ERROR   103                     ;HARD ERR READ SECTOR DX1
3923  016430  000000                        HALT
3924  016432  000776                        BR      .-2                     ;FORCE 250 RESTART
3925
3926  016434  004537  016650          5$:   JSR     R5,EBUFF        ;EMPTY BUFFER INTO D1BUF
3927  016440  002674                           D1BUF
3928
3929  016442  004737  016712                JSR     PC,DATCHK       ;COMPARE D0BUF WITH D1BUF
3930
3931  016446  005237  016530                INC     SECCNT                  ;TOTAL DONE
3932
3933  016452  023737  016530  016526        CMP     SECCNT,MAXSEC           ;DID TOTAL # SECTORS?
3934  016460  001415                        BEQ     8$                      ;BR IF YES
3935
3936  016462  023727  016536  000032        CMP     SECT,#32                ;ELSE DID WE DO LAST SECTOR ON TRACK?
3937  016470  001403                        BEQ     6$                      ;BR IF YES
3938  016472  005237  016536                INC     SECT                    ;ELSE BUMP & DO ANOTHER
3939  016476  000672                        BR      1$
3940
3941  016500  005237  016534          6$:   INC     TRACK                   ;BUMP TRK
3942  016504  012737  000001  016536        MOV     #1,SECT                 ;INIT SECTOR
3943  016512  000664                        BR      1$                      ;& DO ANOTHER
3944
3945  016514  104401  020122          8$:   TYPE    ,COPDUN                 ;DONE MSG
3946  016520  000000                        HALT
3947  016522  000137  016220                JMP     COPY                    ;DO AGAIN IF 'P' TYPED
3948
3949  016526  000000                MAXSEC: 0                               ;TOTAL # SECTORS TO READ FROM DX0
3950  016530  000000                SECCNT: 0                               ;TOTAL # SECTORS WRITTEN TO DX1
3951  016532  000000                ERFLG:  0
3952  016534  000000                TRACK:  0
3953  016536  000000                SECT:   0
```

```
3954
3955                                    ;;*****************************************************************
3956                                    ;ROUTINE TO CALCULATE THE MAX # OF SECTORS TO READ FROM DX0
3957                                    ;;*****************************************************************
3958
3959  016540  012700  026616           GETSEC: MOV     #LSTAD,R0               ;GET MAX ADDR
3960  016544  006200                           ASR     R0                      ;GET WORD CT
3961  016546  005500                           ADC     R0                      ;DONT LOOSE ANY
3962  016550  062700  000377                   ADD     #377,R0
3963  016554  042700  000377                   BIC     #377,R0                 ;ROUND OFF TO NEAREST WHOLE WORD
3964
3965  016560  000300                           SWAB    R0
3966  016562  006300                           ASL     R0
3967  016564  006300                           ASL     R0                      ;DIVIDE BY 100(8) FOR SECTOR CT.
3968
3969  016566  062700  000074                   ADD     #<15.*4>,R0             ;ADD SECTOR CT FOR BLOCKS 0 - 14 OF DX0
3970  016572  062700  000010                   ADD     #8.,R0                  ;2 MORE BLOCKS (NECESSARY FUDGE FACTOR)
3971
3972  016576  010037  016526                   MOV     R0,MAXSEC               ;SAVE
3973                                                                           ;THIS IS THE MAX SECTORS WE WILL READ OFF DX0
3974                                                                           ;& WRITE TO DX1 STARTING AT TRK 1, SECT 1
3975  016602  005037  016530                   CLR     SECCNT                  ;COUNT OF TOTAL SECTORS WRITTEN TO DX1
3976  016606  000207                           RTS     PC
3977
3978
3979                                    ;;*****************************************************************
3980                                    ;ROUTINE TO READ OF WRITE DX0 OR DX1
3981                                    ;R5 POINTS TO THE COMMAND & DISK #
3982                                    ;;*****************************************************************
3983
3984  016610  013700  016534           RDWR:   MOV     TRACK,R0
3985  016614  000300                           SWAB    R0
3986  016616  053700  016536                   BIS     SECT,R0
3987  016622  010037  177174                   MOV     R0,RXSA                 ;LOAD TRK/SEC
3988  016626  012537  177170                   MOV     (R5)+,RXCS              ;ISSUE COMMAND
3989  016632  004537  011442                   JSR     R5,TIMER2               ;WAIT FOR DONE
3990  016636  177170                               RXCS
3991  016640  000200                               DONE
3992  016642  104075                           ERROR   75                      ;NO DONE
3993  016644  000000                           HALT
3994  016646  000205                           RTS     R5
3995
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO EMPTY BUFFER INTO D0BUF OR D1BUF
                                ;R5 POINTS TO IT
                                ;;*****************************************************************

016650  012700  000100        EBUFF:  MOV     #100,R0                 ;WD CT
016654  012501                        MOV     (R5)+,R1                ;BUFFER ADDR
016656  012737  000003  177170        MOV     #EBUF,RXCS              ;ISSUE EMPTY BUFFER COMMAND
016664  013721  177172        1$:     MOV     RXDB,(R1)+              ;PUT IT IN TABLE
016670  005300                        DEC     R0
016672  001374                        BNE     1$
016674  004537  011442                JSR     R5,TIMER2               ;WAIT FOR DONE
016700  177170                            RXCS
016702  000200                            DONE
016704  104075                        ERROR   75                      ;NO DONE
016706  000000                        HALT
016710  000205                        RTS     R5

                                ;;*****************************************************************
                                ;ROUTINE TO COMPARE D0BUF WITH D1BUF
                                ;;*****************************************************************

016712  012700  000100        DATCHK: MOV     #100,R0                 ;WD CT
016716  012701  002424                MOV     #D0BUF,R1
016722  012702  002674                MOV     #D1BUF,R2
016726  022122                1$:     CMP     (R1)+,(R2)+             ;COMPARE?
016730  001403                        BEQ     2$                      ;BR IF YES
016732  104102                        ERROR   102                     ;MISCOMPARE
016734  000000                        HALT
016736  000776                        BR      .-2

016740  005300                2$:     DEC     R0
016742  001371                        BNE     1$
016744  000207                        RTS     PC
```

```
                                        .SBTTL  PROGRAM MESSAGES

                                        ;;*****************************************************************
                                        ; MESSAGES & ERROR FORMATS
                                        ;;*****************************************************************

016746  041600  052514  052123  CLOPT:  .ASCIZ  <CRLF>/CLUSTER OPTION/
016754  051105  047440  052120
016762  047511  000116
016766  052200  051105  020115  CLT1:   .ASCIZ  <CRLF>/TERM #1/
016774  030443    000
016777    200   042524  046522  CLT2:   .ASCIZ  <CRLF>/TERM #2/
017004  021440  000062
017010  052200  051105  020115  CLT3:   .ASCIZ  <CRLF>/TERM #3/
017016  031443    000
017021    200   054523  041516  SCOM:   .ASCIZ  <CRLF>/SYNC COMM/
017026  041440  046517  000115
017034  040600  054523  041516  ACOM:   .ASCIZ  <CRLF>/ASYNC COMM/
017042  041440  046517  000115
017050  050200  044522  052116  PRINT:  .ASCIZ  <CRLF>/PRINTER/
017056  051105    000
017061    040   047520  052122  PORT:   .ASCIZ  / PORT TESTED/
017066  052040  051505  042524
017074  000104
017076  042200  030130    000   DRV0:   .ASCIZ  <CRLF>/DX0/
017103    200   054104  000061  DRV1:   .ASCIZ  <CRLF>/DX1/
017110  041600  047514  045503  CLOCK:  .ASCIZ  <CRLF>/CLOCK/
017116    000
017117    200   045464    000   M4K:    .ASCIZ  <CRLF>/4K/
017123    200   045470    000   M8K:    .ASCIZ  <CRLF>/8K/
017127    200   031061  000113  M12K:   .ASCIZ  <CRLF>/12K/
017134  030600  045466    000   M16K:   .ASCIZ  <CRLF>/16K/
017141    200   030062  000113  M20K:   .ASCIZ  <CRLF>/20K/
017146  031200  045464    000   M24K:   .ASCIZ  <CRLF>/24K/
017153    200   030063  000113  M30K:   .ASCIZ  <CRLF>/30K/
017160  046440  046505  051117  MEM:    .ASCIZ  / MEMORY PRESENT/
017166  020131  051120  051505
017174  047105  000124

017200  047040  052117  050040  NOTPRES:        .ASCIZ  / NOT PRESENT/
017206  042522  042523  052116
017214    000
017215    200   044412  051516  WARNING:        .ASCII  <CRLF><LF>/INSERT SCRATCH DISKS, TYPE 'P' FOR NORMAL TESTING/
017222  051105  020124  041523
017230  040522  041524  020110
017236  044504  045523  026123
017244  052040  050131  020105
017252  050047  020047  047506
017260  020122  047516  046522
017266  046101  052040  051505
017274  044524  043516
017300  023600  032062  043460          .ASCII  <CRLF>/'240G' FOR NORMAL RESTARTS/
```

```
 4087  017306  020047  047506  020122
 4088  017314  047516  046522  046101
 4089  017322  051040  051505  040524
 4090  017330  052122     123
 4091  017333     200  031047  030065          .ASCII  <CRLF>/'250G' TO COPY SYS EXERCISER DISK/
 4092  017340  023507  052040  020117
 4093  017346  047503  054520  051440
 4094  017354  051531  042440  042530
 4095  017362  041522  051511  051105
 4096  017370  042040  051511     113
 4097  017375     200  031047  030066          .ASCII  <CRLF>/'260G' FOR COMPATABILITY PASS 1: WRITE/
 4098  017402  023507  043040  051117
 4099  017410  041440  046517  040520
 4100  017416  040524  044502  044514
 4101  017424  054524  050040  051501
 4102  017432  020123  035061  053440
 4103  017440  044522  042524
 4104  017444  023600  033462  043460          .ASCIZ  <CRLF>/'270G' FOR COMPATABILITY PASS 2: READ/
 4105  017452  020047  047506  020122
 4106  017460  047503  050115  052101
 4107  017466  041101  046111  052111
 4108  017474  020131  040520  051523
 4109  017502  031040  020072  042522
 4110  017510  042101     000
 4111  017513     040  042524  052123  DROP:   .ASCIZ  / TESTING DROPPED/
 4112  017520  047111  020107  051104
 4113  017526  050117  042520  000104
 4114  017534  051040  047125  044516  RUN:    .ASCIZ  / RUNNING/
 4115  017542  043516     000
 4116  017545     040  045517     000  OK:     .ASCIZ  / OK/
 4117  017551     040  051124  000113  TRK:    .ASCIZ  / TRK/
 4118  017556  042040  052101  020101  PATT:   .ASCIZ  / DATA PATT/
 4119  017564  040520  052124     000
 4120  017571     040  040522  042116  RANDOM: .ASCIZ  / RANDOM SEEKS/
 4121  017576  046517  051440  042505
 4122  017604  051513     000
 4123  017607     200  042515  020115  MEMORY: .ASCIZ  <CRLF>/MEM TESTS/
 4124  017614  042524  052123  000123
 4125  017622  042040  047117  000105  DUN:    .ASCIZ  / DONE/
 4126  017630  005200  047105  020104  ENDPAS: .ASCIZ  <CRLF><LF>/END PASS #/
 4127  017636  040520  051523  021440
 4128  017644     000
 4129  017645     011  047524  040524  MSG1:   .ASCIZ  /       TOTAL ERRORS:          /
 4130  017652  020114  051105  047522
 4131  017660  051522  004472  020040
 4132  017666  020040  020040  000040
 4133  017674  004600  004411  047524  MSG2:   .ASCIZ  <CRLF>/                TOTAL SOFT ERRORS:     /
 4134  017702  040524  020114  047523
 4135  017710  052106  042440  051122
 4136  017716  051117  035123  020040
 4137  017724  020040  000040
 4138  017730  005200  004411  052011  MSG3:   .ASCIZ  <CRLF><LF>/                TOTAL ERRORS THIS PASS:/
 4139  017736  052117  046101  042440
 4140  017744  051122  051117  020123
```

```
4141  017752  044124  051511  050040
4142  017760  051501  035123     000
4143  017765     200  004411  051411  MSG4:   .ASCIZ  <CRLF>/                    SOFT  ERRORS THIS PASS:/
4144  017772  043117  020124  042440
4145  020000  051122  051117  020123
4146  020006  044124  051511  050040
4147  020014  051501  035123     000
4148  020021     200  047503  054520  COPMSG: .ASCII  <CRLF>/COPY DX0 TO DX1/
4149  020026  042040  030130  052040
4150  020034  020117  054104     061
4151  020041     200  047111  042523          .ASCIZ  <CRLF>/INSERT SCRATCH DISK IN DX1, TYPE 'P' TO PROCEED/
4152  020046  052122  051440  051103
4153  020054  052101  044103  042040
4154  020062  051511  020113  047111
4155  020070  042040  030530  020054
4156  020076  054524  042520  023440
4157  020104  023520  052040  020117
4158  020112  051120  041517  042505
4159  020120  000104
4160  020122  042200  047117  027105  COPDUN: .ASCIZ  <CRLF>/DONE...TYPE 'P' TO DO AGAIN OR '240G' FOR NORMAL TESTING/
4161  020130  027056  054524  042520
4162  020136  023440  023520  052040
4163  020144  020117  047504  040440
4164  020152  040507  047111  047440
4165  020160  020122  031047  030064
4166  020166  023507  043040  051117
4167  020174  047040  051117  040515
4168  020202  020114  042524  052123
4169  020210  047111  000107
4170  020214  005200  047503  050115  COMPAT: .ASCII  <CRLF><LF>/COMPATABILITY PASS 1 (WRITE) DONE/
4171  020222  052101  041101  046111
4172  020230  052111  020131  040520
4173  020236  051523  030440  024040
4174  020244  051127  052111  024505
4175  020252  042040  047117     105
4176  020257     200  047504  023440          .ASCIZ  <CRLF>/DO 'P' FOR PASS 2 (READ)/
4177  020264  023520  043040  051117
4178  020272  050040  051501  020123
4179  020300  020062  051050  040505
4180  020306  024504     000
4181  020311     200  042504  046526  DEVM:   .ASCIZ  <CRLF>/DEVM = /
4182  020316  036440  000040
4183  020322  042515  047515  054522  EM1:    .ASCIZ  /MEMORY TIMEOUT/
4184  020330  052040  046511  047505
4185  020336  052125     000
4186  020341     115  046505  042040  EM2:    .ASCIZ  /MEM DATA COMP ERR/
4187  020346  052101  020101  047503
4188  020354  050115  042440  051122
4189  020362     000
4190  020363     123  047131  020103  EM3:    .ASCIZ  /SYNC COMM DID NOT RECEIVE SYNC CHAR/
4191  020370  047503  046515  042040
4192  020376  042111  047040  052117
4193  020404  051040  041505  044505
4194  020412  042526  051440  047131
```

```
 4195  020420  020103  044103  051101
 4196  020426     000
 4197  020427     120  044522  052116  EM4:    .ASCIZ  /PRINTER STATUS ERR/
 4198  020434  051105  051440  040524
 4199  020442  052524  020123  051105
 4200  020450  000122
 4201  020452  042524  046522  021440  EM5:    .ASCIZ  /TERM #1 DATA COMP ERR/
 4202  020460  020061  040504  040524
 4203  020466  041440  046517  020120
 4204  020474  051105  000122
 4205  020500  042524  046522  021440  EM6:    .ASCIZ  /TERM #2 DATA COMP ERR/
 4206  020506  020062  040504  040524
 4207  020514  041440  046517  020120
 4208  020522  051105  000122
 4209  020526  042524  046522  021440  EM7:    .ASCIZ  /TERM #3 DATA COMP ERR/
 4210  020534  020063  040504  040524
 4211  020542  041440  046517  020120
 4212  020550  051105  000122
 4213  020554  051501  047131  020103  EM8:    .ASCIZ  /ASYNC COMM DATA COMP ERR/
 4214  020562  047503  046515  042040
 4215  020570  052101  020101  047503
 4216  020576  050115  042440  051122
 4217  020604     000
 4218  020605     123  047131  020103  EM9:    .ASCIZ  /SYNC COMM DATA COMP ERR/
 4219  020612  047503  046515  042040
 4220  020620  052101  020101  047503
 4221  020626  050115  042440  051122
 4222  020634     000
 4223  020635     125  042516  050130  EM10:   .ASCIZ  /UNEXP DISK INTERRUPT/
 4224  020642  042040  051511  020113
 4225  020650  047111  042524  051122
 4226  020656  050125  000124
 4227  020662  054104  020060  051127  EM11:   .ASCIZ  /DX0 WRITE ERR/
 4228  020670  052111  020105  051105
 4229  020676  000122
 4230  020700  054104  020060  040510  EM13:   .ASCIZ  /DX0 HARD ERR - READ SECT/
 4231  020706  042122  042440  051122
 4232  020714  026440  051040  040505
 4233  020722  020104  042523  052103
 4234  020730     000
 4235  020731     104  030130  051440  EM14:   .ASCIZ  /DX0 SOFT ERR - READ SECT/
 4236  020736  043117  020124  051105
 4237  020744  020122  020055  042522
 4238  020752  042101  051440  041505
 4239  020760  000124
 4240  020762  054104  020060  040510  EM15:   .ASCIZ  /DX0 HARD ERR - DATA COMP/
 4241  020770  042122  042440  051122
 4242  020776  026440  042040  052101
 4243  021004  020101  047503  050115
 4244  021012     000
 4245  021013     104  030130  051440  EM16:   .ASCIZ  /DX0 SOFT ERR - DATA COMP/
 4246  021020  043117  020124  051105
 4247  021026  020122  020055  040504
 4248  021034  040524  041440  046517
```

```
 4249  021042  000120
 4250  021044  054104  020061  040510  EM17:   .ASCIZ  /DX1 HARD ERR - WRITE SECT/
 4251  021052  042122  042440  051122
 4252  021060  026440  053440  044522
 4253  021066  042524  051440  041505
 4254  021074  000124
 4255  021076  054104  020061  047523  EM18:   .ASCIZ  /DX1 SOFT ERR - WRITE SECT/
 4256  021104  052106  042440  051122
 4257  021112  026440  053440  044522
 4258  021120  042524  051440  041505
 4259  021126  000124
 4260  021130  054104  020061  040510  EM19:   .ASCIZ  /DX1 HARD ERR - READ SECT/
 4261  021136  042122  042440  051122
 4262  021144  026440  051040  040505
 4263  021152  020104  042523  052103
 4264  021160     000
 4265  021161     104  030530  051440  EM20:   .ASCIZ  /DX1 SOFT ERR - READ SECT/
 4266  021166  043117  020124  051105
 4267  021174  020122  020055  042522
 4268  021202  042101  051440  041505
 4269  021210  000124
 4270  021212  054104  020061  040510  EM21:   .ASCIZ  /DX1 HARD ERR - DATA COMP/
 4271  021220  042122  042440  051122
 4272  021226  026440  042040  052101
 4273  021234  020101  047503  050115
 4274  021242     000
 4275  021243     104  030530  051440  EM22:   .ASCIZ  /DX1 SOFT ERR - DATA COMP/
 4276  021250  043117  020124  051105
 4277  021256  020122  020055  040504
 4278  021264  040524  041440  046517
 4279  021272  000120
 4280  021274  046103  020113  052510  EM23:   .ASCIZ  /CLK HUNG/
 4281  021302  043516     000
 4282  021305     120  044522  052116  EM24:   .ASCIZ  /PRINTER HUNG/
 4283  021312  051105  044040  047125
 4284  021320  000107
 4285  021322  042524  046522  021440  EM25:   .ASCIZ  /TERM #1 HUNG/
 4286  021330  020061  052510  043516
 4287  021336     000
 4288  021337     124  051105  020115  EM26:   .ASCIZ  /TERM #2 HUNG/
 4289  021344  031043  044040  047125
 4290  021352  000107
 4291  021354  042524  046522  021440  EM27:   .ASCIZ  /TERM #3 HUNG/
 4292  021362  020063  052510  043516
 4293  021370     000
 4294  021371     123  047131  020103  EM28:   .ASCIZ  /SYNC COMM HUNG/
 4295  021376  047503  046515  044040
 4296  021404  047125  000107
 4297  021410  051501  047131  020103  EM29:   .ASCIZ  /ASYNC COMM HUNG/
 4298  021416  047503  046515  044040
 4299  021424  047125  000107
 4300  021430  054104  020060  052510  EM30:   .ASCIZ  /DX0 HUNG/
 4301  021436  043516     000
 4302  021441     104  030530  044040  EM31:   .ASCIZ  /DX1 HUNG/
```

```
 4303  021446  047125  000107
 4304  021452  054523  041516  041440  EM32:   .ASCIZ  /SYNC COMM - DATA NOT AVAIL NOT SET/
 4305  021460  046517  020115  020055
 4306  021466  040504  040524  047040
 4307  021474  052117  040440  040526
 4308  021502  046111  047040  052117
 4309  021510  051440  052105     000
 4310  021515     123  047131  020103  EM33:   .ASCIZ  /SYNC COMM - RECVR ACTIVE NOT SET/
 4311  021522  047503  046515  026440
 4312  021530  051040  041505  051126
 4313  021536  040440  052103  053111
 4314  021544  020105  047516  020124
 4315  021552  042523  000124
 4316  021556  054523  041516  041440  EM34:   .ASCIZ  /SYNC COMM - RECVR DONE NOT SET/
 4317  021564  046517  020115  020055
 4318  021572  042522  053103  020122
 4319  021600  047504  042516  047040
 4320  021606  052117  051440  052105
 4321  021614     000
 4322  021615     104  030130  023440  EM35:   .ASCIZ  /DX0 'DEL DATA' BIT 5 NOT SET IN RXES/
 4323  021622  042504  020114  040504
 4324  021630  040524  020047  044502
 4325  021636  020124  020065  047516
 4326  021644  020124  042523  020124
 4327  021652  047111  051040  042530
 4328  021660  000123
 4329  021662  054104  020060  047111  EM36:   .ASCIZ  /DX0 INIT CMD DID NOT READ TRK 1, SEC 1/
 4330  021670  052111  041440  042115
 4331  021676  042040  042111  047040
 4332  021704  052117  051040  040505
 4333  021712  020104  051124  020113
 4334  021720  026061  051440  041505
 4335  021726  030440     000
 4336  021731     104  030130  051040  EM37:   .ASCIZ  /DX0 RESTORE CMD ERR/
 4337  021736  051505  047524  042522
 4338  021744  041440  042115  042440
 4339  021752  051122     000
 4340  021755     104  030130  023440  EM38:   .ASCIZ  /DX0 'INIT DONE' BIT 0 NOT SET IN RXES/
 4341  021762  047111  052111  042040
 4342  021770  047117  023505  041040
 4343  021776  052111  030040  047040
 4344  022004  052117  051440  052105
 4345  022012  044440  020116  054122
 4346  022020  051505     000
 4347  022023     104  030130  051040  EM39:   .ASCIZ  /DX0 READ STATUS CMD ERR/
 4348  022030  040505  020104  052123
 4349  022036  052101  051525  041440
 4350  022044  042115  042440  051122
 4351  022052     000
 4352  022053     104  030130  044440  EM40:   .ASCIZ  /DX0 INV ADDR BIT 1 NOT SET IN RXES/
 4353  022060  053116  040440  042104
 4354  022066  020122  044502  020124
 4355  022074  020061  047516  020124
 4356  022102  042523  020124  047111
```

```
 4357  022110  051040  042530  000123
 4358  022116  054104  020060  051105  EM41:   .ASCIZ  /DX0 ERR BIT NOT SET IN RXCS/
 4359  022124  020122  044502  020124
 4360  022132  047516  020124  042523
 4361  022140  020124  047111  051040
 4362  022146  041530  000123
 4363  022152  054104  020060  047111  EM42:   .ASCIZ  /DX0 INIT CMD ERR/
 4364  022160  052111  041440  042115
 4365  022166  042440  051122     000
 4366  022173     104  030130  051440  EM43:   .ASCIZ  /DX0 SEEK ERR - WRITE SECT/
 4367  022200  042505  020113  051105
 4368  022206  020122  020055  051127
 4369  022214  052111  020105  042523
 4370  022222  052103     000
 4371  022225     104  030130  044040  EM45:   .ASCIZ  /DX0 HARD SEEK ERR - READ SECT/
 4372  022232  051101  020104  042523
 4373  022240  045505  042440  051122
 4374  022246  026440  051040  040505
 4375  022254  020104  042523  052103
 4376  022262     000
 4377  022263     104  030130  051440  EM46:   .ASCIZ  /DX0 SOFT SEEK ERR - READ SECT/
 4378  022270  043117  020124  042523
 4379  022276  045505  042440  051122
 4380  022304  026440  051040  040505
 4381  022312  020104  042523  052103
 4382  022320     000
 4383  022321     104  030530  044040  EM47:   .ASCIZ  /DX1 HARD SEEK ERR - WRITE SECT/
 4384  022326  051101  020104  042523
 4385  022334  045505  042440  051122
 4386  022342  026440  053440  044522
 4387  022350  042524  051440  041505
 4388  022356  000124
 4389  022360  054104  020061  047523  EM48:   .ASCIZ  /DX1 SOFT SEEK ERR - WRITE SECT/
 4390  022366  052106  051440  042505
 4391  022374  020113  051105  020122
 4392  022402  020055  051127  052111
 4393  022410  020105  042523  052103
 4394  022416     000
 4395  022417     104  030530  044040  EM49:   .ASCIZ  /DX1 HARD SEEK ERR - READ SECT/
 4396  022424  051101  020104  042523
 4397  022432  045505  042440  051122
 4398  022440  026440  051040  040505
 4399  022446  020104  042523  052103
 4400  022454     000
 4401  022455     104  030530  051440  EM50:   .ASCIZ  /DX1 SOFT SEEK ERR - READ SECT/
 4402  022462  043117  020124  042523
 4403  022470  045505  042440  051122
 4404  022476  026440  051040  040505
 4405  022504  020104  042523  052103
 4406  022512     000
 4407  022513     104  030530  051040  EM51:   .ASCIZ  /DX1 RESTORE CMD ERR/
 4408  022520  051505  047524  042522
 4409  022526  041440  042115  042440
 4410  022534  051122     000
```

```
 4411  022537     104  051511  020113  EM52:   .ASCIZ  /DISK 'DONE' NOT SET/
 4412  022544  042047  047117  023505
 4413  022552  047040  052117  051440
 4414  022560  052105     000
 4415  022563     106  046111  020114  EM53:   .ASCIZ  /FILL & EMPTY BUFF TEST... DATA MISCOMP/
 4416  022570  020046  046505  052120
 4417  022576  020131  052502  043106
 4418  022604  052040  051505  027124
 4419  022612  027056  042040  052101
 4420  022620  020101  044515  041523
 4421  022626  046517  000120
 4422  022632  040504  040524  046440  EM54:   .ASCIZ  /DATA MISCOMP/
 4423  022640  051511  047503  050115
 4424  022646     000
 4425  022647     122  050105  040514  EM55:   .ASCIZ  /REPLACE DX0...10 BAD SECTORS/
 4426  022654  042503  042040  030130
 4427  022662  027056  030456  020060
 4428  022670  040502  020104  042523
 4429  022676  052103  051117  000123
 4430  022704  042522  046120  041501  EM56:   .ASCIZ  /REPLACE DX1...10 BAD SECTORS/
 4431  022712  020105  054104  027061
 4432  022720  027056  030061  041040
 4433  022726  042101  051440  041505
 4434  022734  047524  051522     000
 4435  022741     104  030130  044040  EM57:   .ASCIZ  /DX0 HARD CRC ERR/
 4436  022746  051101  020104  051103
 4437  022754  020103  051105  000122
 4438  022762  054104  020060  040504  EM58:   .ASCIZ  /DX0 DATA CRC ERR/
 4439  022770  040524  041440  041522
 4440  022776  042440  051122     000
 4441  023003     104  030130  050040  EM59:   .ASCIZ  /DX0 PREV CRC ERR WITH DATA OK/
 4442  023010  042522  020126  051103
 4443  023016  020103  051105  020122
 4444  023024  044527  044124  042040
 4445  023032  052101  020101  045517
 4446  023040     000
 4447  023041     104  030530  044040  EM60:   .ASCIZ  /DX1 HARD CRC ERR/
 4448  023046  051101  020104  051103
 4449  023054  020103  051105  000122
 4450  023062  054104  020061  040504  EM61:   .ASCIZ  /DX1 DATA CRC ERR/
 4451  023070  040524  041440  041522
 4452  023076  042440  051122     000
 4453  023103     104  030530  050040  EM62:   .ASCIZ  /DX1 PREV CRC ERR WITH DATA OK/
 4454  023110  042522  020126  051103
 4455  023116  020103  051105  020122
 4456  023124  044527  044124  042040
 4457  023132  052101  020101  045517
 4458  023140     000
 4459
```

```
 4460  023141     105  051122  051117  DH1:    .ASCIZ  /ERROR #       ERR PC/
 4461  023146  021440  042411  051122
 4462  023154  050040  000103
 4463  023160  051105  047522  020122  DH2:    .ASCIZ  /ERROR #       ERR PC  ADDR/
 4464  023166  004443  051105  020122
 4465  023174  041520  040411  042104
 4466  023202  000122
 4467  023204  051105  047522  020122  DH3:    .ASCIZ  /ERROR #       ERR PC  ADDR    EXPECT  RECVD/
 4468  023212  004443  051105  020122
 4469  023220  041520  040411  042104
 4470  023226  004522  054105  042520
 4471  023234  052103  020040  042522
 4472  023242  053103  000104
 4473  023246  051105  047522  020122  DH4:    .ASCIZ  /ERROR #       ERR PC  PR STATUS/
 4474  023254  004443  051105  020122
 4475  023262  041520  050011  020122
 4476  023270  052123  052101  051525
 4477  023276     000
 4478  023277     105  051122  051117  DH5:    .ASCIZ  /ERROR #       ERR PC  EXPECT  RECVD/
 4479  023304  021440  042411  051122
 4480  023312  050040  004503  054105
 4481  023320  042520  052103  020040
 4482  023326  042522  053103  000104
 4483  023334  051105  047522  020122  DH6:    .ASCIZ  /ERROR #       DXTST#  ERR PC  RXCS    RXES    RXSA/
 4484  023342  004443  054104  051524
 4485  023350  021524  042411  051122
 4486  023356  050040  004503  054122
 4487  023364  051503  051011  042530
 4488  023372  004523  054122  040523
 4489  023400     000
 4490  023401     105  051122  051117  DH7:    .ASCIZ  /ERROR #       DXTST#  ERR PC  RXCS    RXES    TRACK   SECTOR  # RETRIE
 4491  023406  021440  042011  052130
 4492  023414  052123  004443  051105
 4493  023422  020122  041520  051011
 4494  023430  041530  004523  054122
 4495  023436  051505  052011  040522
 4496  023444  045503  051411  041505
 4497  023452  047524  004522  020043
 4498  023460  042522  051124  042511
 4499  023466  000123
 4500  023470  054104  051524  020124  DH8:    .ASCIZ  /DXTST #       ERR PC  RXCS    RXES    RXSA    EXPECT  RECVD  # RETRIES
 4501  023476  004443  051105  020122
 4502  023504  041520  051011  041530
 4503  023512  004523  054122  051505
 4504  023520  051011  051530  004501
 4505  023526  054105  042520  052103
 4506  023534  020040  042522  053103
 4507  023542  020104  021440  051040
 4508  023550  052105  044522  051505
 4509  023556     000
 4510
 4511          023560                  .EVEN
```

```
4512
4513  023560  026050  001116  000000  DT1:    .WORD   ERRNUM,$ERRPC,0
4514  023566  026050  001116  004436  DT2:    .WORD   ERRNUM,$ERRPC,ADDR,0
4515  023574  000000
4516  023576  026050  001116  004436  DT3:    .WORD   ERRNUM,$ERRPC,ADDR,PAT,MEMHLD,0
4517  023604  004434  004432  000000
4518  023612  026050  001116  012144  DT4:    .WORD   ERRNUM,$ERRPC,SPRS,0
4519  023620  000000
4520  023622  026050  001116  012244  DT5:    .WORD   ERRNUM,$ERRPC,T1CHR,T1HLD,0
4521  023630  012246  000000
4522  023634  026050  001116  012344  DT6:    .WORD   ERRNUM,$ERRPC,T2CHR,T2HLD,0
4523  023642  012346  000000
4524  023646  026050  001116  012444  DT7:    .WORD   ERRNUM,$ERRPC,T3CHR,T3HLD,0
4525  023654  012446  000000
4526  023660  026050  001116  012544  DT8:    .WORD   ERRNUM,$ERRPC,COMCHR,COMHLD,0
4527  023666  012546  000000
4528  023672  026050  003120  001116  DT9:    .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,SRXSA,0
4529  023700  013050  013052  013054
4530  023706  000000
4531  023710  026050  003120  001116  DT10:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D0TRK,D0SEC,D0ERR,0
4532  023716  013050  013052  002406
4533  023724  002410  002404  000000
4534  023732  003120  001116  177170  DT11:   .WORD   DXTST,$ERRPC,RXCS,RXES,RXSA,D0EXP,D0REC,D0CERR,0
4535  023740  177172  177174  002416
4536  023746  002420  002414  000000
4537  023754  026050  003120  001116  DT12:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D1TRK,D1SEC,D1ERR,0
4538  023762  013050  013052  002656
4539  023770  002660  002654  000000
4540  023776  003120  001116  177170  DT13:   .WORD   DXTST,$ERRPC,RXCS,RXES,RXSA,D1EXP,D1REC,D1CERR,0
4541  024004  177172  177174  002666
4542  024012  002670  002664  000000
4543  024020  026050  003120  001116  DT14:   .WORD   ERRNUM,DXTST,$ERRPC,RXCS,RXES,RXSA,0
4544  024026  177170  177172  177174
4545  024034  000000
4546  024036  026050  001116  006274  DT15:   .WORD   ERRNUM,$ERRPC,EXP0,EBHLD,0
4547  024044  006272  000000
4548  024050  026050  001116  006276  DT16:   .WORD   ERRNUM,$ERRPC,EXP1,EBHLD,0
4549  024056  006272  000000
4550
4551  024062  000000                  DF:     .WORD   0               ;OCTAL FORMAT FOR ALL
4552  024064  000000                          .WORD   0
4553  024066  000000                          .WORD   0
4554
4555  024070  000000                  DF1:    .WORD   0               ;OCTAL,OCTAL
4556  024072  000000                          .WORD   0               ;OCTAL,OCTAL
4557  024074  000400                          .WORD   400             ;DECIMAL,OCTAL
4558  024076  000401                          .WORD   401             ;DECIMAL,DECIMAL
4559
4560  024100  000000                  DF2:    .WORD   0               ;OCTAL,OCTAL
4561  024102  000000                          .WORD   0               ;OCTAL,OCTAL
4562  024104  000000                          .WORD   0               ;OCTAL,OCTAL
4563  024106  000400                          .WORD   400             ;DECIMAL,OCTAL
4564
```

```
                                        .SBTTL  TYPE ROUTINE
                                        ;;*****************************************************************
                                        ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
                                        ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
                                        ;*NOTE1:       $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
                                        ;*NOTE2:       $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
                                        ;*NOTE3:       $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
                                        ;*
                                        ;*CALL:
                                        ;*1) USING A TRAP INSTRUCTION
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*

024110  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
024114  100002                          BPL     1$              ;;BR IF YES
024116  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
024120  000430                          BR      3$              ;;LEAVE
024122  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
024124  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
024130  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
024136  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
024140  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
024146  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
024150  010037  024160                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
024154  004737  025242                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
024160  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
024162  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
024170  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
024172  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
024174  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
024176  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
024200  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
024202  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
024206  000002                          RTI                     ;;RETURN
024210  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
024214  001430                          BEQ     8$
024216  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
024222  001006                          BNE     5$
024224  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
024226  104401                          TYPE                    ;;TYPE A CR AND LF
024230  001165                          $CRLF
024232  105037  024366                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
024236  000755                          BR      2$              ;;GET NEXT CHARACTER
024240  004737  024322          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
024244  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
024250  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
024252  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
024256  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
024262  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
```

```
 4619  024264  004737  024322                 JSR     PC,$TYPEC       ;;GO TYPE A NULL
 4620  024270  105337  024366                 DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
 4621  024274  000770                         BR      7$              ;;LOOP
 4622
 4623                                 ;HORIZONTAL TAB PROCESSOR
 4624
 4625  024276  112716  000040         8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 4626  024302  004737  024322         9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 4627  024306  132737  000007  024366         BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 4628  024314  001372                         BNE     9$              ;;TAB STOP
 4629  024316  005726                         TST     (SP)+           ;;POP SPACE OFF STACK
 4630  024320  000724                         BR      2$              ;;GET NEXT CHARACTER
 4631  024322  105777  154622         $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 4632  024326  100375                         BPL     $TYPEC
 4633  024330  116677  000002  154614         MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 4634  024336  122766  000015  000002         CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 4635  024344  001003                         BNE     1$              ;;BRANCH IF NO
 4636  024346  105037  024366                 CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 4637  024352  000406                         BR      $TYPEX          ;;EXIT
 4638  024354  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 4639  024362  001402                         BEQ     $TYPEX          ;;BRANCH IF YES
 4640  024364  105227                         INCB    (PC)+           ;;COUNT THE CHARACTER
 4641  024366  000000         $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 4642  024370  000207         $TYPEX: RTS     PC
 4643
```

```
                                        .SBTTL  TTY INPUT ROUTINE
                                        ;;*********************************************************************
                                        .ENABL  LSB
                                        ;;*********************************************************************
                                        ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
                                        ;*ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL
                                        ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL
                                        ;*WHEN OPERATING IN TTY FLAG MODE.
024372  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR          ;;IS THE SOFT-SWR SELECTED?
024400  001114                         BNE     15$                 ;;BRANCH IF NO
024402  105777  154536                 TSTB    @$TKS               ;;CHAR THERE?
024406  100111                         BPL     15$                 ;;IF NO, DON'T WAIT AROUND
024410  117746  154532                 MOVB    @$TKB,-(SP)         ;;SAVE THE CHAR
024414  042716  177600                 BIC     #^C177,(SP)         ;;STRIP-OFF THE ASCII
024420  022726  000007                 CMP     #7,(SP)+            ;;IS IT A CONTROL G?
024424  001102                         BNE     15$                 ;;NO, RETURN TO USER
024426  123727  001134  000001         CMPB    $AUTOB,#1           ;;ARE WE RUNNING IN AUTO-MODE?
024434  001476                         BEQ     15$                 ;;BRANCH IF YES
024436  104401  025204                 TYPE    ,$CNTLG             ;;ECHO THE CONTROL-G (^G)
024442  104401  025211         $GTSWR: TYPE    ,$MSWR              ;;TYPE CURRENT CONTENTS
024446  013746  000176                 MOV     SWREG,-(SP)         ;;SAVE SWREG FOR TYPEOUT
024452  104402                         TYPOC                       ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
024454  104401  025222                 TYPE    ,$MNEW              ;;PROMPT FOR NEW SWR
024460  005046                 19$:    CLR     -(SP)               ;;CLEAR COUNTER
024462  005046                         CLR     -(SP)               ;;THE NEW SWR
024464  105777  154454         7$:     TSTB    @$TKS               ;;CHAR THERE?
024470  100375                         BPL     7$                  ;;IF NOT TRY AGAIN
024472  117746  154450                 MOVB    @$TKB,-(SP)         ;;PICK UP CHAR
024476  042716  177600                 BIC     #^C177,(SP)         ;;MAKE IT 7-BIT ASCII
024502  021627  000003                 CMP     (SP),#3             ;;IS IT A CONTROL-C?
024506  001015                         BNE     9$                  ;;BRANCH IF NOT
024510  104401  025172                 TYPE    ,$CNTLC             ;;YES, ECHO CONTROL-C (^C)
024514  062706  000006                 ADD     #6,SP               ;;CLEAN UP STACK
024520  123727  001135  000001         CMPB    $INTAG,#1           ;;REENABLE TTY KEYBOARD INTERRUPTS?
024526  001003                         BNE     8$                  ;;BRANCH IF NO
024530  012777  000100  154406         MOV     #100,@$TKS          ;;ALLOW TTY KEYBOARD INTERRUPTS
024536  000137  010760         8$:     JMP     GT$DEV              ;;CONTROL-C RESTART
024542  021627  000025         9$:     CMP     (SP),#25            ;;IS IT A CONTROL-U?
024546  001005                         BNE     10$                 ;;BRANCH IF NOT
024550  104401  025177                 TYPE    ,$CNTLU             ;;YES, ECHO CONTROL-U (^U)
024554  062706  000006         20$:    ADD     #6,SP               ;;IGNORE PREVIOUS INPUT
024560  000737                         BR      19$                 ;;LET'S TRY IT AGAIN
024562  021627  000015         10$:    CMP     (SP),#15            ;;IS IT A <CR>?
024566  001022                         BNE     16$                 ;;BRANCH IF NO
024570  005766  000004                 TST     4(SP)               ;;YES, IS IT THE FIRST CHAR?
```

```
 4698  024574  001403                        BEQ     11$             ;;BRANCH IF YES
 4699  024576  016677  000002  154334        MOV     2(SP),@SWR      ;;SAVE NEW SWR
 4700  024604  062706  000006          11$:  ADD     #6,SP           ;;CLEAR UP STACK
 4701  024610  104401  001165          14$:  TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 4702  024614  123727  001135  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 4703  024622  001003                        BNE     15$             ;;BRANCH IF NOT
 4704  024624  012777  000100  154312        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 4705  024632  000002          15$:  RTI                             ;;RETURN
 4706  024634  004737  024322  16$:  JSR     PC,$TYPEC               ;;ECHO CHAR
 4707  024640  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
 4708  024644  002420                        BLT     18$             ;;BRANCH IF YES
 4709  024646  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
 4710  024652  003015                        BGT     18$             ;;BRANCH IF YES
 4711  024654  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 4712  024660  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4713  024664  001403                        BEQ     17$             ;;BRANCH IF YES
 4714  024666  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
 4715  024670  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
 4716  024672  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
 4717  024674  005266  000002          17$:  INC     2(SP)           ;;KEEP COUNT OF CHAR
 4718  024700  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4719  024704  000667                        BR      7$              ;;GET THE NEXT ONE
 4720  024706  104401  001164          18$:  TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 4721  024712  000720                        BR      20$             ;;SIMULATE CONTROL-U
 4722                                .DSABL  LSB
 4723
 4724
 4725                                ;;**************************************************************************
 4726                                ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4727                                ;*CALL:
 4728                                ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 4729                                ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4730                                ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4731                                ;
 4732
 4733  024714  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 4734  024716  016666  000004  000002        MOV     4(SP),2(SP)     ;;SAVE THE PS
 4735  024724  105777  154214          1$:   TSTB    @$TKS           ;;WAIT FOR
 4736  024730  100375                        BPL     1$              ;;A CHARACTER
 4737  024732  117766  154210  000004        MOVB    @$TKB,4(SP)     ;;READ THE TTY
 4738  024740  042766  177600  000004        BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4739  024746  026627  000004  000023        CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4740  024754  001013                        BNE     3$              ;;BRANCH IF NO
 4741  024756  105777  154162          2$:   TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 4742  024762  100375                        BPL     2$              ;;LOOP UNTIL ITS THERE
 4743  024764  117746  154156                MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 4744  024770  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4745  024774  022627  000021                CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4746  025000  001366                        BNE     2$              ;;IF NOT DISCARD IT
 4747  025002  000750                        BR      1$              ;;YES, RESUME
 4748  025004  026627  000004  000140  3$:   CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4749  025012  002407                        BLT     4$              ;;BRANCH IF YES
 4750  025014  026627  000004  000175        CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4751  025022  003003                        BGT     4$              ;;BRANCH IF YES
```

```
4752  025024  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
4753  025032  000002                  4$:     RTI                     ;;GO BACK TO USER
4754                                  ;;**************************************************************
4755                                  ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
4756                                  ;*CALL:
4757                                  ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
4758                                  ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
4759                                  ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
4760
4761  025034  010346                  $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
4762  025036  012703  025162          1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
4763  025042  022703  025172          2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
4764  025046  101415                          BLOS    4$              ;;BR IF YES
4765  025050  104410                          RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
4766  025052  112613                          MOVB    (SP)+,(R3)      ;;GET CHARACTER
4767  025054  122713  000003                  CMPB    #3,(R3)         ;;IS IT A CONTROL-C?
4768  025060  001005                          BNE     10$             ;;BRANCH IF NO
4769  025062  104401  025172                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
4770  025066  012603                          MOV     (SP)+,R3        ;;RESTORE R3
4771  025070  000137  010760                  JMP     GT$DEV          ;;GOTO CONTROL-C RESTART
4772  025074  122713  000177          10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
4773  025100  001003                          BNE     3$              ;;SKIP IF NOT
4774  025102  104401  001164          4$:     TYPE    ,$QUES          ;;TYPE A '?'
4775  025106  000753                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
4776  025110  111337  025160          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
4777  025114  104401  025160                  TYPE    ,9$
4778  025120  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
4779  025124  001346                          BNE     2$              ;;LOOP IF NOT RETURN
4780  025126  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
4781  025132  104401  001166                  TYPE    ,$LF            ;;TYPE A LINE FEED
4782  025136  012603                          MOV     (SP)+,R3        ;;RESTORE R3
4783  025140  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
4784  025142  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
4785  025150  012766  025162  000004          MOV     #$TTYIN,4(SP)
4786  025156  000002                          RTI                     ;;RETURN
4787  025160     000                  9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
4788  025161     000                          .BYTE   0               ;;TERMINATOR
4789  025162  000010                  $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
4790  025172  041536  005015     000 $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL ''C''
4791  025177     136  006525  000012 $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL ''U''
4792  025204  043536  005015     000 $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL ''G''
4793  025211     015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR = /
4794  025216  036440  000040
4795  025222  020040  042516  020127 $MNEW:  .ASCIZ  /  NEW = /
4796  025230  020075     000
4797          025234                  .EVEN
```

```
                                        .SBTTL  APT COMMUNICATIONS ROUTINE
                                        ;
                                        ;;*************************************************************************
025234  112737  000001  025500  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
025242  112737  000001  025476  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
025250  000403                          BR      $ATYC
025252  112737  000001  025500  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
025260                          $ATYC:
025260  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
025262  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
025264  105737  025476                  TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
025270  001450                          BEQ     5$              ;;IF NOT:  BR
025272  122737  000001  001210          CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
025300  001031                          BNE     3$              ;;IF NOT:  BR
025302  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
025310  001425                          BEQ     3$              ;;IF NOT:  BR
025312  017600  000004                  MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
025316  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
025324  005737  001170          1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
025330  001375                          BNE     1$              ;;IF NOT:  WAIT
025332  010037  001204                  MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
025336  105720                  2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
025340  001376                          BNE     2$
025342  163700  001204                  SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
025346  006200                          ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
025350  010037  001206                  MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
025354  012737  000004  001170          MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
025362  000413                          BR      5$
025364  017637  000004  025410  3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
025372  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
025400  013746  177776                  MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
025404  004737  024110                  JSR     PC,$TYPE        ;;CALL TYPE MACRO
025410  000000                  4$:     .WORD   0
025412                          5$:
025412  105737  025500          10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
025416  001416                          BEQ     12$             ;;IF NOT:  BR
025420  005737  001210                  TST     $ENV            ;;RUNNING UNDER APT?
025424  001413                          BEQ     12$             ;;IF NOT:  BR
025426  005737  001170          11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
025432  001375                          BNE     11$             ;;IF NOT:  WAIT
025434  017637  000004  001172          MOV     @4(SP),$FATAL   ;;GET ERROR #
025442  062766  000002  000004          ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
025450  005237  001170                  INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
025454  105037  025500          12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
025460  105037  025477                  CLRB    $LFLG           ;;CLEAR LOG FLAG
025464  105037  025476                  CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
025470  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025472  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025474  000207                          RTS     PC              ;;RETURN
025476     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
025477     000          $LFLG:  .BYTE   0               ;;LOG FLAG
025500     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
        025502                          .EVEN
        000200          APTSIZE=200
```

```
 4852          000001                  APTENV=001
 4853          000100                  APTSPOOL=100
 4854          000040                  APTCSUP=040
```

```
                        .SBTTL  ERROR HANDLER ROUTINE

                        ;;*****************************************************************
                        ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                        ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                        ;*AND GO TO MYTYPE ON ERROR
                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                        ;*SW15=1         HALT ON ERROR
                        ;*SW13=1         INHIBIT ERROR TYPEOUTS
                        ;*SW10=1         BELL ON ERROR
                        ;*CALL
                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

025502                  $ERROR:
025502  104407                  CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
025504  105237  001103  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
025510  001775          BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
025512  013777  001102  153422  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
025520  032777  002000  153412  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
025526  001402          BEQ     1$                      ;;NO - SKIP
025530  104401  001160  TYPE    ,$BELL                  ;;RING BELL
025534  005237  001112  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
025540  011637  001116          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
025544  162737  000002  001116  SUB     #2,$ERRPC
025552  117737  153340  001114  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
025560  032777  020000  153352  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
025566  001004          BNE     20$                     ;;SKIP TYPEOUTS
025570  004737  026016          JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
025574  104401  001165          TYPE    ,$CRLF
025600                  20$:
025600  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
025606  001007          BNE     2$                      ;;NO,SKIP APT ERROR REPORT
025610  113737  001114  025622          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
025616  004737  025252          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
025622     000          21$:    .BYTE   0
025623     000                  .BYTE   0
025624  000777          22$:    BR      22$             ;;APT ERROR LOOP
025626  005777  153306  2$:     TST     @SWR            ;;HALT ON ERROR
025632  100002          BPL     3$                      ;;SKIP IF CONTINUE
025634  000000          HALT                            ;;HALT ON ERROR!
025636  104407          CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
025640                  3$:
025640  000002          RTI                             ;;RETURN
```

```
                                .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

                                ;;*****************************************************************
                                ;*THIS ROUTINE USES THE "ITEM CONTROL BYTE" ($ITEMB) TO DETERMINE WHICH
                                ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" ($ERRTB),
                                ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

025642                          $ERRTYP:
025642  104401  001165                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025646  010046                        MOV     R0,-(SP)        ;;SAVE R0
025650  005000                        CLR     R0              ;;PICKUP THE ITEM INDEX
025652  153700  001114                BISB    @#$ITEMB,R0
025656  001004                        BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                                ;;TYPE THE PC OF THE ERROR
025660  013746  001116                MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;ERROR ADDRESS
025664  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025666  000445                        BR      10$             ;;GET OUT
025670  005300                1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
025672  006300                        ASL     R0              ;;      WORK FOR THE ERROR TABLE
025674  006300                        ASL     R0
025676  006300                        ASL     R0
025700  062700  001250                ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
025704  012037  025714                MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
025710  001404                        BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
025712  104401                        TYPE                    ;;TYPE THE "ERROR MESSAGE"
025714  000000                2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
025716  104401  001165                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025722  012037  025732        3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
025726  001404                        BEQ     5$              ;;SKIP TYPEOUT IF 0
025730  104401                        TYPE                    ;;TYPE THE "DATA HEADER"
025732  000000                4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
025734  104401  001165                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
025740  010146                5$:     MOV     R1,-(SP)        ;;SAVE R1
025742  012001                        MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
025744  001415                        BEQ     9$              ;;BR IF NO DATA TO BE TYPED
025746  012000                        MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
025750  105720                6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
025752  001003                        BNE     7$              ;;BR IF DECIMAL
025754  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
025756  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025760  000402                        BR      8$
025762                          7$:
025762  013146                        MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
025764  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
025766  005711                8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
025770  001403                        BEQ     9$              ;;BR IF NO
025772  104401  026012                TYPE    ,11$            ;;TYPE TWO(2) SPACES
025776  000764                        BR      6$              ;;LOOP

026000  012601                9$:     MOV     (SP)+,R1        ;;RESTORE R1
026002  012600                10$:    MOV     (SP)+,R0        ;;RESTORE R0
026004  104401  001165                TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
026010  000207                        RTS     PC              ;;RETURN
```

```
4952  026012  020040  000          11$:    .ASCIZ  / /          ;;TWO(2) SPACES
4953          026016                       .EVEN
4954
4955
4956
4957  026016  113737  001114  026050  MYTYPE: MOVB    $ITEMB,ERRNUM   ;FOR ERROR TYPEOUT
4958  026024  013737  013054  001174          MOV     SRXSA,$TESTN    ;FOR APT
4959                                                                  ;APT PRINTS $TESTN & $FATAL IN THAT ORDER.
4960                                                                  ;$TESTN WILL BE FLOPPY RXSA
4961                                                                  ;$FATAL IS ERR #
4962  026032  004737  025642                  JSR     PC,$ERRTYP      ;TYPE ERR MSG
4963  026036  005237  026052                  INC     TERR
4964  026042  005237  026056                  INC     PERR
4965  026046  000207                          RTS     PC
4966
4967  026050  000000          ERRNUM: 0                               ;FOR ERR TYPEOUT
4968
4969  026052  000000          TERR:   0                               ;TOTAL ERROR COUNT
4970  026054  000000          TSERR:  0                               ;TOTAL SOFT ERR COUNT
4971  026056  000000          PERR:   0                               ;PASS ERR COUNT
4972  026060  000000          PSERR:  0                               ;PASS SOFT ERR COUNT... ALL FOR EOP TYPEOUT
```

```
                                .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE

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

 026062                         $TYPDS:
 026062  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 026064  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 026066  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 026070  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 026072  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 026074  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 026100  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 026104  100004                         BPL     1$              ;;BR IF INPUT IS POS.
 026106  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 026110  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 026116  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 026120  012703  026276                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 026124  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 026130  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 026132  016001  026266                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 026136  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 026140  002402                         BLT     4$              ;;BR IF DONE
 026142  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 026144  000774                         BR      3$
 026146  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 026150  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
 026152  001002                         BNE     5$              ;;FALL THROUGH IF 0
 026154  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 026156  100407                         BMI     7$              ;;BR IF YES
 026160  106316                 5$:     ASLB    (SP)            ;;MSD?
 026162  103003                         BCC     6$              ;;BR IF NO
 026164  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 026172  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 026176  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 026202  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
 026204  005720                         TST     (R0)+           ;;JUST INCREMENTING
 026206  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
 026212  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
 026214  003002                         BGT     8$              ;;GO TO EXIT
 026216  010502                         MOV     R5,R2           ;;GET THE LSD
 026220  000764                         BR      6$              ;;GO CHANGE TO ASCII
 026222  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
 026224  100003                         BPL     9$              ;;BR IF NO
 026226  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
 026234  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
 026236  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
```

```
 5027  026240  012603                        MOV     (SP)+,R3        ;;POP STACK INTO R3
 5028  026242  012602                        MOV     (SP)+,R2        ;;POP STACK INTO R2
 5029  026244  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
 5030  026246  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
 5031  026250  104401  026276                TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 5032  026254  016666  000002  000004        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 5033  026262  012616                        MOV     (SP)+,(SP)
 5034  026264  000002                        RTI                     ;;RETURN TO USER
 5035  026266  023420                $DTBL:  10000.
 5036  026270  001750                        1000.
 5037  026272  000144                        100.
 5038  026274  000012                        10.
 5039  026276  000004                $DBLK:  .BLKW   4
```

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

026306  017646  000000          $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
026312  116637  000001  026531          MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
026320  112637  026533                  MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
026324  062716  000002                  ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
026330  000406                          BR      $TYPON
026332  112737  000001  026531  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
026340  112737  000006  026533          MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
026346  112737  000005  026530  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
026354  010346                          MOV     R3,-(SP)        ;;SAVE R3
026356  010446                          MOV     R4,-(SP)        ;;SAVE R4
026360  010546                          MOV     R5,-(SP)        ;;SAVE R5
026362  113704  026533                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
026366  005404                          NEG     R4
026370  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
026374  110437  026532                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
026400  113704  026531                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
026404  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
026410  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
026412  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
026414  000404                          BR      3$              ;;GO DO MSB
026416  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
026420  006105                          ROL     R5
026422  006105                          ROL     R5
026424  010503                          MOV     R5,R3
026426  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
026430  105337  026532                  DECB    $OMODE          ;;TYPE THIS DIGIT?
026434  100016                          BPL     7$              ;;BR IF NO
026436  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
026442  001002                          BNE     4$              ;;TEST FOR 0
```

```
 5094  026444  005704                        TST     R4              ;;SUPPRESS THIS 0?
 5095  026446  001403                        BEQ     5$              ;;BR IF YES
 5096  026450  005204                4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 5097  026452  052703  000060              BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 5098  026456  052703  000040        5$:   BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 5099  026462  110337  026526              MOVB    R3,8$           ;;SAVE FOR TYPING
 5100  026466  104401  026526              TYPE    ,8$             ;;GO TYPE THIS DIGIT
 5101  026472  105337  026530        7$:   DECB    $OCNT           ;;COUNT BY 1
 5102  026476  003347                        BGT     2$              ;;BR IF MORE TO DO
 5103  026500  002402                        BLT     6$              ;;BR IF DONE
 5104  026502  005204                        INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 5105  026504  000744                        BR      2$              ;;GO DO THE LAST DIGIT
 5106  026506  012605                6$:     MOV     (SP)+,R5        ;;RESTORE R5
 5107  026510  012604                        MOV     (SP)+,R4        ;;RESTORE R4
 5108  026512  012603                        MOV     (SP)+,R3        ;;RESTORE R3
 5109  026514  016666  000002  000004       MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 5110  026522  012616                        MOV     (SP)+,(SP)
 5111  026524  000002                        RTI                     ;;RETURN
 5112  026526     000                8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 5113  026527     000                        .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 5114  026530     000             $OCNT:     .BYTE   0               ;;OCTAL DIGIT COUNTER
 5115  026531     000             $OFILL:    .BYTE   0               ;;ZERO FILL SWITCH
 5116  026532  000000             $OMODE:    .WORD   0               ;;NUMBER OF DIGITS TO TYPE
```

```
                                .SBTTL  TRAP DECODER

                                ;;**********************************************************************
                                ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE ''TRAP'' INSTRUCTION
                                ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
                                ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
                                ;*GO TO THAT ROUTINE.

026534  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
026536  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
026542  005740                          TST     -(R0)           ;;BACKUP BY 2
026544  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
026546  006300                          ASL     R0              ;;POSITION FOR INDEXING
026550  016000  026570                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
026554  000200                          RTS     R0              ;;GO TO ROUTINE

                                ;;THIS IS USE TO HANDLE THE ''GETPRI'' MACRO

026556  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
026560  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
026566  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

                                ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
                                ;*BY THE ''TRAP'' INSTRUCTION.

                                ;       ROUTINE
                                ;       -------
026570  026556                  $TRPAD: .WORD   $TRAP2
026572  024110                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
026574  026332                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
026576  026306                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
026600  026346                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
026602  026062                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

026604  024442                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

026606  024372                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
026610  024714                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
026612  025034                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
026614  010760                          GT$DEV  ;;CALL=GTDEVM   TRAP+12(104412) ^C WILL OPEN UP $DEVM

026616                          LSTAD:
                                        ;MAX ADDR FOR 8K WITH APT = 37400
                                        ;MAX ADDR FOR 8K NO APT BUT TO SAVE ABS LOADER = 37500
        003174                  .END    START
```

```
ABASE = 176500          664#   1146   1187
ACDW1 = 000000         1146
ACDW2 = 000000         1146
ACOM    017034         1915   2582   4051#
ACPUOP= 000000         1146   1161
ADDR    004436         1665*  1679*  1684*  1712*  1717*  1745#  4514   4516
ADDW0 = 000000         1146
ADDW1 = 000000         1146
ADDW10= 000000         1146
ADDW11= 000000         1146
ADDW12= 000000         1146
ADDW13= 000000         1146
ADDW14= 000000         1146
ADDW15= 000000         1146
ADDW2 = 000000         1146
ADDW3 = 000000         1146
ADDW4 = 000000         1146
ADDW5 = 000000         1146
ADDW6 = 000000         1146
ADDW7 = 000000         1146
ADDW8 = 000000         1146
ADDW9 = 000000         1146
ADEVCT= 000000         1146   1152
ADEVM = 000017          664#  1146   1188
AENV  = 000000         1146   1157
AENVM = 000000         1146   1158
AFATAL= 000000         1146   1149
AMADR1= 000000         1146   1174
AMADR2= 000000         1146   1178
AMADR3= 000000         1146   1181
AMADR4= 000000         1146   1184
AMAMS1= 000000         1146   1168
AMAMS2= 000000         1146   1176
AMAMS3= 000000         1146   1179
AMAMS4= 000000         1146   1182
AMOD  = 030000          878#  1892   1894
AMRB  = 176612          837#  1903   3083
AMRC  = 176610          836#  1588*  1589*  1619*  1779*  1780*  1895*  1904*  1928*  1929*
AMRSRV  012466         1899   3083#
AMRVEC= 000330          810#  1898
AMSGAD= 000000         1146   1154
AMSGLG= 000000         1146   1155
AMSGTY= 000000         1146   1148
AMTST   005222         1822   1868   1871   1887#
AMTYP1= 000000         1146   1169
AMTYP2= 000000         1146   1177
AMTYP3= 000000         1146   1180
AMTYP4= 000000         1146   1183
AMXB  = 176616          839#  3080*
AMXC  = 176614          838#  1905*  3079*  3095*
AMXSRV  012450         1900   3079#
AMXVEC= 000334          811#
APASS = 000000         1146   1151
APRIOR= 000000         1146
```

```
APTCSU= 000040   4595  4854#
APTENV= 000001   4588  4810  4852#  4885
APTSIZ= 000200   1551  4851#
APTSPO= 000100   4590  4812  4853#
ASWREG= 000000   1146  1159
ATESTN= 000000   1146  1150
AUNIT = 000000   1146  1153
AUSWR = 000000    664#  1146  1160
AVECT1= 000300    664#  1146  1185
AVECT2= 000000   1146  1186
BITHLD  011524   2816*  2820  2834#
BIT0  = 000001    783#   951  1047  1934
BIT00 = 000001    773#   783
BIT01 = 000002    772#   782
BIT02 = 000004    771#   781
BIT03 = 000010    770#   780
BIT04 = 000020    769#   779
BIT05 = 000040    768#   778
BIT06 = 000100    767#   777
BIT07 = 000200    766#   776
BIT08 = 000400    765#   775
BIT09 = 001000    764#   774
BIT1  = 000002    782#   916   938   974  1046  1968
BIT10 = 002000    763#   931   965  1820  4873
BIT11 = 004000    762#   964  1005  1887
BIT12 = 010000    761#   930   945   963   980  1004  1732  1765  1807  1867  1913  1931  1956
                 1990  2024  3454  3468  3487  3815  3829  3849
BIT13 = 020000    760#   929   944   962  1965  4880
BIT14 = 040000    759#   928   943   961   979   987  1999
BIT15 = 100000    758#   927   942   960   978  1003  1022  1774  3575  3754
BIT2  = 000004    781#   937   973  1045  2002
BIT3  = 000010    780#   936   972  1011  1044  1830  1890
BIT4  = 000020    779#   971  1010  1025  1026  1043
BIT5  = 000040    778#   935   970  1009  1042
BIT6  = 000100    777#   934   950   969  1008  1024
BIT7  = 000200    776#   933   949   968  1007  1023  1041  1752  2562
BIT8  = 000400    775#   967  1006  2040  2134  2172  2213  2263  2287  2332  2398  2435
BIT9  = 001000    774#   932   966  2042  2153  2180  2238  2268  2361
BPTVEC= 000014    790#
BREAK = 000001    951#
B110  = 001000    885#
B1200 = 003400    890#
B134  = 001400    886#
B150  = 002000    887#
B1800 = 004000    891#
B19200= 007400    898#
B2000 = 004400    892#
B2400 = 005000    893#  1936  1938  1970  1972  2004  2006
B300  = 002400    888#
B3600 = 005400    894#
B4800 = 006000    895#
B50   = 000000    883#
B600  = 003000    889#
B7200 = 006400    896#
```

```
B75    = 000400            884#
B9600  = 007000            897#    1787    1892    1894
CDET   = 010000            930#     963#
CHAR5  = 000000            909#
CHAR6  = 000004            910#
CHAR7  = 000010            911#
CHAR8  = 000014            912#    1787    1892    1894    1936    1938    1970    1972    2004    2006
CHR5   = 000000            991#
CHR6   = 002000            992#
CHR7   = 004000            993#
CHR8   = 006000            994#    1833
CKSWR  = 104407           2650     4869    4895    5156#
CK0BAD  011616            2226     2881#
CK1BAD  011730            2251     2929#
CLK    = 177546            831#    1613*   1757*
CLKTST  004442            1733     1736    1752#
CLKVEC= 000100             808#    1754*   1755*
CLOCK   017110            1767     2578    4060#
CLOPT   016746            2555     4040#
CLPRES  010606            1922     2551*   2554*   2599#
CLRBAD  011526            1627     2841#
CLT1    016766            1958     2583    4043#
CLT2    016777            1992     2584    4045#
CLT3    017010            2026     2585    4047#
CL1TST  005362            1888     1911    1914    1922#
CL2TST  005552            1932     1954    1957    1965#
CL3TST  005704            1966     1988    1991    1999#
COMCHR  012544            1856*    1861    1902*   1908    3080    3084    3088    3090*   3093*   3098#   3106    3107    3130
                          3134     3136*   3139*   4526
COMHLD  012546            1845*    1846    1903*   3083*   3084    3099#   3127*   3128    3130    4526
COMPAT  020214            2502     4170#
COMP1   003132            1489#    1495*   1499*   1503*   1507*   1577    1629    2188    2500    2504*   3571    3750
COMP2   003134            1490#    1498*   1504*   1508*   1579    1631    2140    2159    2273    2505*   3573    3752
CONSOL= 010000             873#
COPDUN  020122            3945     4160#
COPFLG  003130            1488#    1502*   1509*   1574    1638    3873*
COPMSG  020021            3874     4148#
COPY    016220            1640     3873#   3947
CR     = 000015             698#    1792    2972    2985    4634    4644
CRC    = 000010            1044#    3321    3695
CRLF   = 000200             699#    1573    4040    4043    4045    4047    4049    4051    4053    4058    4059    4060    4062
                           4063     4064    4065    4066    4067    4068    4077    4086    4091    4097    4104    4123    4126
                           4133     4138    4143    4148    4151    4160    4170    4176    4181    4605    4644
CTHLD   011242            2727*    2731    2745#
CTREG   011332            2755*    2759*   2770#   2783*   2787*
CTREG1  011334            2754*    2762*   2771#   2782*   2790*
CTS    = 020000             929#     962#
CT1    = 000000             874#    1936    1938
CT2    = 050000             875#    1970    1972
CT3    = 060000             876#    2004    2006
DATCHK  016712            3929     4020#
DD     = 000040            1042#    3344
DDISP  = 177570             705#    1129    1539
DEVM    020311            2657     4181#
```

```
DF      024062  1208  1211  1214  1217  1220  1223  1226  1229  1232  1235  1238  1241  1244
                1283  1286  1289  1292  1295  1298  1301  1304  1307  1310  1313  1316  1319
                1322  1325  1328  1331  1337  1340  1343  1346  1349  1352  1355  1358  1385
                1388  1391  1394  1400  1403  1406  1409  4551#
DF1     024070  1247  1250  1253  1256  1265  1268  1271  1274  1361  1364  1367  1370  1373
                1376  1379  1382  1424  1433  4555#
DF2     024100  1259  1262  1277  1280  1334  1427  1436  4560#
DH1     023141  1223  1283  1289  1292  1295  1298  1301  1310  1313  1316  4460#
DH2     023160  1208  1211  1217  4463#
DH3     023204  1214  1220  4467#
DH4     023246  1226  1286  4473#
DH5     023277  1229  1232  1235  1238  1241  1391  1394  4478#
DH6     023334  1244  1304  1307  1319  1322  1325  1328  1331  1337  1340  1343  1346  1349
                1352  1355  1358  1385  1388  1400  1403  1406  1409  4483#
DH7     023401  1247  1250  1253  1256  1265  1268  1271  1274  1361  1364  1367  1370  1373
                1376  1379  1382  1424  1433  4490#
DH8     023470  1259  1262  1277  1280  1334  1427  1436  4500#
DISPLA  001142  1129#  1539*  1547*  4872*
DISPRE  000174  1056#  1547
DNA   = 100000  1003#
DNAIE = 000040  1009#
DONE  = 000200   933#   968#  1023#  1841  2047  2055  2069  2083  3991  4010
DROP    017513  2575  4111#
DRV0    017076  2579  3456  3470  3489  4058#
DRV1    017103  2580  3817  3831  3851  4059#
DSC   = 100000   960#
DSI   = 100000   927#
DSIE  = 000040   935#   970#
DSRDY = 001000   932#   966#
DSWR  = 177570   704#  1128  1538
DTR   = 000002   938#   974#  1588  1589  1619  1779  1780  1834  1895  1928  1929
DT1     023560  1223  1283  1289  1292  1295  1298  1301  1310  1313  1316  4513#
DT10    023710  1247  1250  1253  1256  1361  1364  1367  1370  1424  4531#
DT11    023732  1259  1262  1334  1427  4534#
DT12    023754  1265  1268  1271  1274  1373  1376  1379  1382  1433  4537#
DT13    023776  1277  1280  1436  4540#
DT14    024020  1304  1307  1319  1322  1325  1328  1343  1349  1388  1400  1403  1406  1409
                4543#
DT15    024036  1391  4546#
DT16    024050  1394  4548#
DT2     023566  1208  1211  1217  4514#
DT3     023576  1214  1220  4516#
DT4     023612  1226  1286  4518#
DT5     023622  1229  4520#
DT6     023634  1232  4522#
DT7     023646  1235  4524#
DT8     023660  1238  1241  4526#
DT9     023672  1244  1331  1337  1340  1346  1352  1355  1358  1385  4528#
DUN     017622  1735  1870  3460  3472  3491  3821  3833  3853  4125#
DXTST   003120  1476#  2038*  2120*  2203*  2290*  2335*  2401*  2439*  3216  3342  3404  3428  3499
                3501  3860  4528  4531  4534  4537  4540  4543
DXTST0  006036  1924  2000  2022  2025  2038#
DXTST1  006300  1636  2043  2100  2120#
DXTST2  006634  2189  2203#
```

```
DXTST3  007170          2274    2287#
DXTST4  007344          2288    2301    2332#
DXTST5  007616          2333    2362    2370    2398#
DXTST6  007734          2399    2412    2421    2435#
DX1   = 000020          1025#   2166    2257    2366    2382    3526    3556    3578    3594    3614    3649    3662    3683
                        3692    3731    3835    3854    3902    3915
DOBAD   002624          1456#   2841    2843    2859    2866    2888    2889
DOBUF   002424          1455#   2075    2088    2098    3361    3382    3899    4021
DOCDAT  013750          3362    3377#
DOCERR  002414          1451#   2124*   2206*   2292*   2312*   2352*   2403*   3402    3417    3423*   3427*   4534
DOCMER  002412          1450#   3268*   3377*   3387*   3395    3444*   3482*
DOCNT   002402          1446#   2207*   2265    3484*   3485
DOCRC   002422          1454#   3323*   3327*   3397    3412    3424*
DODISP  013040          2142*   2145*   2229*   2294*   2316*   2336*   2354*   2405*   2414*   2444*   3175    3176#   3196*
                        3215*   3239*   3274*   3288*   3308*   3317*   3362*   3512*   3521*   3533*   3541*
DODUN   014612          3432    3503    3512    3521    3533    3541    3551#
DOEBUF  013712          2316    3348    3360#
DOERR   002404          1447#   2123*   2205*   2291*   2311*   2351*   2402*   3233    3246    3256*   3267*   3271*   3303
                        3305*   3331    3341*   3443*   3481*   4531
DOEXP   002416          1452#   3385*   4534
DOFBUF  013056          2145    2405    3192#   3264    3274
DOINIT  014476          2294    3510#
DOINV   014574          2444    3541#
DONEXT  014454          3317    3329    3425    3499#
DONXT1  014174          3443#   3500
DONXT2  014370          3481#   3502
DOPAT   002400          1445#   2122*   2174    3473*
DORCHK  013466          3288    3301#
DOREC   002420          1453#   3386*   4534
DOREST  014522          2336    3521#
DORSEC  013432          2142    2229    2354    3269    3283#   3307    3308    3430    3452
DOSEC   002410          1449#   2126*   2224*   2314*   2349*   2441*   2467*   2886    3195    3213    3257    3259*   3261
                        3263*   3266*   3285    3381    3445    3447*   3449    3451*   3465*   4531
DOSTAT  014556          2414    3533#
DOTMO   011336          2176    2753*   2764*   2772#
DOTRK   002406          1448#   2125*   2219*   2313*   2348*   2440*   2466*   2884    3193    3211    3283    3379    3458
                        3462    3464*   4531
DOWCHK  013176          3215    3231#
DOWSEC  013122          3196    3211#
D1BAD   003074          1474#   2846    2848    2907    2914    2936    2937
D1BUF   002674          1473#   3729    3756    3927    4022
D1CDAT  015512          3730    3744#
D1CERR  002664          1469#   2130*   2210*   2379*   3774    3783    3789*   3793*   4540
D1CMER  002662          1468#   3642*   3744*   3761*   3768    3805*   3844*
D1CNT   002652          1464#   2211*   2270    3846*   3847
D1CRC   002672          1472#   3697*   3701*   3770    3779    3790*
D1DISP  013046          2161*   2164*   2254*   2364*   2381*   3178    3179#   3525*   3577*   3593*   3613*   3648*   3661*
                        3682*   3691*   3730*
D1DUN   014626          3525    3555#
D1EBUF  015454          3716    3728#
D1ERR   002654          1465#   2129*   2209*   2378*   3607    3620    3630*   3641*   3645*   3677    3679*   3705    3715*
                        3804*   3843*   4537
D1EXP   002666          1470#   3759*   4540
D1FBUF  014642          2164    3566#   3638    3648
```

```
D1INT   013042          3173    3178#
D1NEXT  016206          3691    3703    3791    3860#
D1NXT1  015726          3804#   3861
D1NXT2  016122          3843#   3862
D1PAT   002650          1463#   2128*   2182    3834*
D1RCHK  015252          3661    3675#
D1REC   002670          1471#   3760*   4540
D1REST  014540          2364    3525#
D1RSEC  015216          2161    2254    2381    3643    3656#   3681    3682    3794    3813
D1SEC   002660          1467#   2132*   2249*   2377*   2934    3569    3591    3631    3633*   3635    3637*   3640*   3658
                        3748    3806    3808*   3810    3812*   3826*   4537
D1TMO   011426          2184    2781*   2792*   2798#
D1TRK   002656          1466#   2131*   2244*   2376*   2932    3567    3589    3656    3746    3819    3823    3825*   4537
D1WCHK  014762          3593    3605#
D1WSEC  014726          3577    3589#
EBHLD   006272          2089*   2093*   2094    2104#   4546    4548
EBUF  = 000003          1031#   2076    3363    3731    4004
EBUFF   016650          3898    3926    4002#
EMTVEC= 000030           793#   1530*   1531*
EM1     020322          1208    1211    1217    4183#
EM10    020635          1244    4223#
EM11    020662          1247    1250    4227#
EM13    020700          1253    1400    4230#
EM14    020731          1256    4235#
EM15    020762          1259    4240#
EM16    021013          1262    4245#
EM17    021044          1265    1403    4250#
EM18    021076          1268    4255#
EM19    021130          1271    1409    4260#
EM2     020341          1214    1220    4186#
EM20    021161          1274    4265#
EM21    021212          1277    4270#
EM22    021243          1280    4275#
EM23    021274          1283    4280#
EM24    021305          1286    4282#
EM25    021322          1289    4285#
EM26    021337          1292    4288#
EM27    021354          1295    4291#
EM28    021371          1298    4294#
EM29    021410          1301    4297#
EM3     020363          1223    4190#
EM30    021430          1304    1319    1322    1325    1328    1343    1349    4300#
EM31    021441          1307    4302#
EM32    021452          1310    4304#
EM33    021515          1313    4310#
EM34    021556          1316    4316#
EM35    021615          1331    4322#
EM36    021662          1334    4329#
EM37    021731          1337    4336#
EM38    021755          1340    4340#
EM39    022023          1346    4347#
EM4     020427          1226    4197#
EM40    022053          1352    4352#
EM41    022116          1355    4358#
```

```
EM42    022152            1358    4363#
EM43    022173            1361    1364    4366#
EM45    022225            1367    4371#
EM46    022263            1370    4377#
EM47    022321            1373    4383#
EM48    022360            1376    4389#
EM49    022417            1379    4395#
EM5     020452            1229    4201#
EM50    022455            1382    4401#
EM51    022513            1385    4407#
EM52    022537            1388    4411#
EM53    022563            1391    1394    4415#
EM54    022632            1406    4422#
EM55    022647            1415    4425#
EM56    022704            1418    4430#
EM57    022741            1424    4435#
EM58    022762            1427    4438#
EM59    023003            1430    4441#
EM6     020500            1232    4205#
EM60    023041            1433    4447#
EM61    023062            1436    4450#
EM62    023103            1439    4453#
EM7     020526            1235    4209#
EM8     020554            1238    4213#
EM9     020605            1241    4218#
ENDPAS  017630            2481    4126#
ENVPAR= 001400             998#
EOP     010114            2102    2190    2275    2451    2480#
EPAR  = 000060             904#
ERFLG   016532            3879*   3890*   3891    3897*   3906*   3907    3913*   3919*   3920    3951#
ERR   = 100000             942#    978#   1022#
ERRNUM  026050            4513    4514    4516    4518    4520    4522    4524    4526    4528    4531    4537    4543    4546
                          4548    4957*   4967#
ERRVEC= 000004             786#   1536    1537*   1548*   1654*   1655*   1729*   1730*   2515*   2516*   2558*   2559*
EXAMKB  010744            1652    2648#   2733    2758    2786    2822
EXP0    006274            2105#   4546
EXP1    006276            2106#   4548
FBUF  = 000001            1030#   2061    3197    3578
FIN     003122            1483#   2146*   2165*   2231*   2256*   2296*   2318*   2337*   2353*   2365*   2380*   2407*   2416*
                          2446*   2756    2784    3461*   3483*   3551*   3555*   3822*   3845*
FLGHLD  011240            2726*   2731    2744#
FREE    005220            1824*   1873    1878*   1881#
FRERR = 020000             944#
GETSEC  016540            3877    3959#
GNS   = ****** U          1572    5148    5149    5150    5151    5152    5154    5156    5157    5158    5159
GTDEVM= 104412            1595    5159#
GTSWR = 104406            1567    5154#
GT$DEV  010760            2657#   4685    4771    5159
HFDUP = 000010            1011#
HILIM   010736            2217*   2222*   2242*   2247*   2637    2642#
HINUM   010740            2619    2627    2632*   2644#
HLD42   004440            1517*   1688    1746#
HT    = 000011             696#   4603    4644
ID    = 000001            1047#   2307
```

```
IE     = 000100   934#  950#  969#  1008#  1024#  1613  1757  1794  1805  1857  1858  1904  1905
                 1947  1948  1981  1982   2015   2016  2147  2166  2232  2257  2297  2319  2338
                 2355  2366  2382  2408   2417   2447  2999  3015  3026  3042  3052  3068  3079
                 3095  3105  3113  3140   3144   3197  3218  3220  3240  3275  3289  3309  3318
                 3363  3474  3492  3513   3522   3526  3534  3542  3552  3556  3578  3594  3614
                 3649  3662  3683  3692   3731   3835  3854
INITAL= 000011   1034#  2051  3513
INTFLG  012012   1661*  1663  1675*  1677  1708*  1710  1756*  1760  2517*  2521*  2524  2541*  2544
                 2956*  2958#
INTSRV  012004   1654  1754  2515  2956#
INVADR= 000002   1046#  2453
INVCT   010112   2438*  2462  2465*  2473#
IOTVEC= 000020    791#
LDATA  = 000004  1045#
LD0BAD  011560   2857#  3235  3312
LD1BAD  011672   2905#  3609  3686
LF     = 000012   697#  2974  2981  4077  4126  4138  4170  4638  4644
LINCT   012142   1791*  1798  2986*  2991#
LOLIM   010734   2216*  2221*  2241*  2246*  2635  2641#
LONUM   010742   2618  2625  2631*  2645#
LOOP    003706   1575  1578  1580  1599  1603  1608#  2506
LSTAD   026616   1668  1673  3959  5161#
LSTCHR  012636   1826*  3110  3112*  3121#
MAINT  = 000002   916#  1892  1936  1970  2004
MAXMEM  010630   1657  1703  2532*  2533*  2610#
MAXSEC  016526   3933  3949#  3972*
MAXSK   003126   1486#  1583*  2265  2270  2498*  3485  3847
MAXTRK  003124   1484#  1582*  1635*  2217  2242  2497*  3462  3823
MCLK   = 004000  1005#  1828  1832
MEM     017160   2538  4069#
MEMCT   004430   1671*  1697  1699*  1723*  1724  1738#
MEMHLD  004432   1681*  1682  1714*  1715  1743#  4516
MEMORY  017607   1734  4123#
MEMTBL  010612   2535  2602#
MEMTST  004022   1639  1652#
MM     = 010000  1004#  1832
MR     = 000400  1006#  1827  1866
MSG1    017645   2484  4129#
MSG2    017674   2487  4133#
MSG3    017730   2490  4138#
MSG4    017765   2493  4143#
MYTYPE  026016   4882  4957#
M12K    017127   2604  4064#
M16K    017134   2605  4065#
M20K    017141   2606  4066#
M24K    017146   2607  4067#
M30K    017153   2608  4068#
M4K     017117   2602  4062#
M8K     017123   2603  4063#
NOTPRE  017200   2556  2592  4074#
NULL    010252   2496  2508#
ODDPAR= 001000    999#  1833
OK      017545  4116#
OPAR   = 000020   905#  1892  1894
```

```
ORERR = 040000          943#   979#
PARAM = 177420          829#  1787*  1892*  1894*  1936*  1938*  1970*  1972*  2004*  2006*
PARERR= 010000          945#   980#
PAT     004434         1672*  1676   1682   1700*  1706*  1709   1715   1726*  1744#  4516
PATT    017556         3471   3832   4118#
PC    =%000007          717#  1597*  1610*  1622*  1627*  1652*  2149*  2168*  2218*  2223*  2234*  2243*  2248*
                       2259*  2299*  2321*  2340*  2357*  2368*  2384*  2410*  2419*  2449*  2596*  2639*  2651*
                       2679*  2733*  2758*  2768*  2786*  2796*  2807*  2822*  2851*  2870*  2918*  3235*  3312*
                       3609*  3686*  3877*  3929*  3976*  4031*  4593*  4612*  4619*  4626*  4640*  4642*  4706*
                       4829*  4846*  4882*  4888*  4951*  4962*  4965*
PERR    026056         1624*  2491   4964*  4971#
PIRQ  = 177772          703#
PIRQVE= 000240          797#
PORT    017061         1812   4055#
PRB   = 177516          834#  2971*
PRCHR   012140         1792*  2971   2972   2974   2976   2978*  2981*  2983*  2985*  2990#
PRINT   017050         1809   2586   2591   4053#
PRPORT  012146         1783*  1786*  1803   1810   2993#
PRPRES  010610         1776   2590*  2594*  2600#
PRS   = 177514          833#  1781   1794*  1805*  2588   2965
PRSRV   012014         1788   2965#
PRT   = 020000          877#  1787
PRTST   004534         1753   1763   1766   1774#
PRVEC = 000200          806#  1788*  1789*
PR0   = 000000          720#  1617
PR1   = 000040          721#
PR2   = 000100          722#
PR3   = 000140          723#
PR4   = 000200          724#  1516
PR5   = 000240          725#
PR6   = 000300          726#
PR7   = 000340          727#
PS    = 177776          700#   701
PSERR   026060         1625*  2494   3249*  3334*  3420*  3623*  3708*  3786*  4972#
PSW   = 177776          701#
PWRVEC= 000024          792#
RACT  = 004000          964#
RAND    010632         2218   2223   2243   2248   2618#  2636   2638
RANDOM  017571         3490   3852   4120#
RDCHR = 104410         4765   5157#
RDLIN = 104411         5158#
RDWR    016610         3885   3901   3914   3984#
RDY   = 000200          949#  1007#  1041#
REGHLD  011522         2815*  2820   2833#
RESTOR= 000017         1037#  3240   3275   3309   3318   3522   3526   3614   3649   3683   3692
RESVEC= 000010          787#
RING  = 040000          928#   961#
RNF   = 000020         1043#  3236   3250   3272   3306   3313   3335   3610   3624   3646   3680   3687   3709
RSEC  = 000007         1033#  3289   3662   3886   3915
RSTAT = 000013         1035#  3534
RTS   = 000004          937#   973#  1588   1589   1619   1779   1780   1834   1895   1928   1929
RUN     017534         1768   1814   1916   1959   1993   2027   4114#
RXCS  = 177170          847#  2046   2051*  2054   2061*  2068   2076*  2082   2147*  2166*  2232*  2257*  2297*
                       2303   2319*  2338*  2344   2355*  2366*  2372   2382*  2408*  2417*  2423   2447*  2458
```

```
SMRVEC= 000340   813#  1852
SMTST   004734  1775   1782   1801   1808   1813   1820#
SMXB  = 176626   845#  1837*  3106*
SMXC  = 176624   844#  1827*  1828*  1832*  1836*  1858*  1866*  3105*  3113*  3117*  3144*
SMXSRV  012550  1854   3105#
SMXVEC= 000344   814#
SP    =%000006   716#  1511*  1528*  1536*  1544*  1548   1608*  2482*  2485*  2488*  2491*  2494*  2658*
                2662*  2663*  2667*  2668*  2670   2672   2674   2675*  2680   2682   2684*  2685   2687*
                2688*  2689*  2691*  2692*  2707*  2713   2767*  2795*  2857*  2869   2881*  2882*  2898
                2899   2905*  2917   2929*  2930*  2946   2947   3458*  3819*  4586*  4587   4597*  4599
                4600   4601*  4603   4605   4607   4613   4615*  4617*  4625*  4629   4633   4634   4638
                4658*  4659*  4660   4667*  4670*  4671*  4675*  4676*  4678   4681*  4688   4691*  4695
                4697   4699   4700*  4707   4709   4711*  4712   4714*  4715*  4716*  4717*  4718*  4733*
                4734*  4737*  4738*  4739   4743*  4744*  4745   4748   4750   4752*  4761*  4766   4770
                4782   4783*  4784*  4785*  4806*  4807*  4814   4815*  4826   4827*  4828*  4838   4839*
                4844   4845   4877   4907*  4912*  4931*  4937*  4941*  4948   4949   4986*  4987*  4988*
                4989*  4990*  4991*  4992   4995*  5008   5010*  5012   5022   5024   5026   5027   5028
                5029   5030   5032*  5033*  5065*  5066   5067   5068*  5073*  5074*  5075*  5081   5106
                5107   5108   5109*  5110*  5125*  5126   5136*  5137*
SPRS    012144  2965*  2966   2992#  4518
SREC  = 002000   931#   965#
SRSYN = 000020   971#  1834
SRXCS   013050  3159*  3168*  3182#  4528   4531   4537
SRXES   013052  3160*  3169*  3183#  3236   3250   3272   3306   3313   3321   3335   3610   3624   3646
                3680   3687   3695   3709   4528   4531   4537
SRXSA   013054  3161*  3170*  3184#  4528   4958
STACK = 001100   691#  1511   1528
START   003174  1064   1067   1507#  5164
START1  003136  1072   1495#
START2  003144  1074   1498#
START3  003156  1070   1502#
STKLMT= 177774   702#
STSYN = 000400   967#
SWR     001140  1128#  1526   1538*  1540   1546*  1553*  1565   1732   1765   1807   1867   1913   1956
                1990   2024   3454   3468   3487   3815   3829   3849   4654   4699*  4873   4880   4892
SWREG   000176  1057#  1546   1565   4654   4667
SW0   = 000001   755#
SW00  = 000001   745#   755
SW01  = 000002   744#   754
SW02  = 000004   743#   753
SW03  = 000010   742#   752
SW04  = 000020   741#   751
SW05  = 000040   740#   750
SW06  = 000100   739#   749
SW07  = 000200   738#   748
SW08  = 000400   737#   747
SW09  = 001000   736#   746
SW1   = 000002   754#
SW10  = 002000   735#
SW11  = 004000   734#
SW12  = 010000   733#
SW13  = 020000   732#
SW14  = 040000   731#
SW15  = 100000   730#
```

```
SW2    = 000004            753#
SW3    = 000010            752#
SW4    = 000020            751#
SW5    = 000040            750#
SW6    = 000100            749#
SW7    = 000200            748#
SW8    = 000400            747#
SW9    = 001000            746#
SXMIT  = 000010            936#    972#
SYNC1  = 040000            987#
SYNC2  = 000000            986#
S1     003200             1496    1508#
S2     003204             1500    1509#
S3     003210             1505    1511#
TBITVE= 000014             788#
TERR   026052             1585*   2485    4963*   4969#
TIMD0  011244             2149    2234    2299    2321    2340    2357    2410    2419    2449    2753#
TIMD1  011340             2168    2259    2368    2384    2781#
TIMER  011160             1759    1797    1860    1907    1950    1984    2018    2726#
TIMER1 011430             1610    1622    2804#
TIMER2 011442             1839    2045    2053    2067    2081    2815#   3989    4008
TKVEC  = 000060            795#
TK1B   = 176502            853#    1946    3003
TK1S   = 176500            852#    1947*   2542
TK1SRV 012166             1942    3003#
TK1VEC= 000300             818#    1941
TK2B   = 176512            858#    1980    3030
TK2S   = 176510            857#    1981*
TK2SRV 012266             1976    3030#
TK2VEC= 000310             820#    1975
TK3B   = 176522            863#    2014    3056
TK3S   = 176520            862#    2015*   2546
TK3SRV 012366             2010    3056#
TK3VEC= 000320             822#    2009
TPVEC  = 000064            796#
TP1B   = 176506            855#    3000*
TP1S   = 176504            854#    1948*   2999*   3015*
TP1SRV 012150             1943    2999#
TP1VEC= 000304             819#
TP2B   = 176516            860#    3027*
TP2S   = 176514            859#    1982*   3026*   3042*
TP2SRV 012250             1977    3026#
TP2VEC= 000314             821#
TP3B   = 176526            865#    3053*
TP3S   = 176524            864#    2016*   3052*   3068*
TP3SRV 012350             2011    3052#
TP3VEC= 000324             823#
TRACK  016534             3880*   3941*   3952#   3984
TRAPVE= 000034             794#    1532*   1533*
TRK    017551             3457    3818    4117#
TRTVEC= 000014             789#
TSERR  026054             1586*   2488    3248*   3333*   3419*   3622*   3707*   3785*   4970#
TYPDS  = 104405            2483    2486    2489    2492    2495    3459    3820    4942    5152#
TYPE   = 104401            1559    1604    1734    1735    1767    1768    1809    1812    1814    1869    1870    1915    1916
```

```
                    1958    1959    1992    1993    2026    2027    2481    2484    2487    2490    2493    2496    2502
                    2536    2538    2555    2556    2573    2575    2591    2592    2657    2660    2676    2695    3456
                    3457    3460    3470    3471    3472    3489    3490    3491    3817    3818    3821    3831    3832
                    3833    3851    3852    3853    3874    3945    4608    4665    4666    4669    4680    4690    4701
                    4720    4769    4774    4777    4781    4875    4883    4906    4923    4925    4928    4930    4945
                    4950    5031    5100    5148#
TYPOC = 104402      2659    4668    4914    4938    5149#
TYPON = 104404      5151#
TYPOS = 104403      5150#
T1CHR   012244      1945*   1951    3000    3004    3008    3010*   3013*   3018#   4520
T1HLD   012246      1946*   3003*   3004    3019#   4520
T2CHR   012344      1979*   1985    3027    3031    3035    3037*   3040*   3045#   4522
T2HLD   012346      1980*   3030*   3031    3046#   4522
T3CHR   012444      2013*   2019    3053    3057    3061    3063*   3066*   3071#   4524
T3HLD   012446      2014*   3056*   3057    3072#   4524
ULITE = 040000       879#
USEL  = 000020      1026#   3172
WARNIN  017215      1604    4077#
WDDSEC= 000015      1036#   3218
WSEC  = 000005      1032#   3220    3542    3594    3902
$APTHD  001000      1088    1094#
$ASTAT= ****** U    4832    4847
$ATYC   025260      4803    4805#
$ATY1   025234      4801#
$ATY3   025242      4593    4802#
$ATY4   025252      4804#   4888
$AUTOB  001134      1125#   1569*   4662    4797
$BASE   001244      1187#
$BDADR  001122      1120#
$BDDAT  001126      1122#
$BELL   001160      1138#   4875    4898
$CHARC  024366      4610*   4620*   4627    4636*   4641#
$CKSWR  024372      4654#   5156
$CMTAG  001100      1108#   1523    1524    1534
$CM3  = 000000      1138#
$CNTLC  025172      4680    4769    4790#
$CNTLG  025204      4665    4792#
$CNTLU  025177      4690    4791#
$CPUOP  001216      1161#
$CRLF   001165      1140#   2676    4609    4644    4701    4790    4883    4898    4906    4925    4930    4950
$DBLK   026276      4997    5031    5039#
$DEVCT  001200      1152#   2736*   2761*   2789*   2825*   2987*   3014*   3041*   3067*   3094*   3114*   3118*   3141*
$DEVM   001246      1188#   1600    1602    1752    1774    1820    1830    1887    1890    1926    1931    1934    1965
                    1968    1999    2002    2040    2042    2134    2153    2172    2180    2213    2238    2263    2268
                    2287    2332    2361    2398    2435    2564    2658    2674*
$DTBL   026266      5000    5035#
$ENV    001210      1157#   1518    1563    1593    1598    2648    4588    4810    4834    4885
$ENVM   001211      1158#   1551    4590    4595    4812
$ERFLG  001103      1111#   4870*   4898
$ERMAX  001115      1117#
$ERROR  025502      1530    4868#
$ERRPC  001116      1118#   4513    4514    4516    4518    4520    4522    4524    4526    4528    4531    4534    4537
                    4540    4543    4546    4548    4877*   4878*   4879    4898    4912
$ERRTB  001250      1205#   4920
```

```
$ERRTY  025642          4905#   4962
$ERTTL  001112          1115#   4876*   4898
$ETABL  001210          1156#
$ETEND  001250          1100    1189#
$FATAL  001172          1149#   4838*
$FFLG   025500          4801*   4804*   4832    4841*   4849#
$FILLC  001156          1136#   4613    4644
$FILLS  001155          1135#   4644
$GDADR  001120          1119#
$GDDAT  001124          1121#
$GTSWR  024442          4666#   5154
$HD   = 000000           675
$HIBTS  001000          1095#
$ICNT   001104          1112#
$INTAG  001135          1126#   4682    4702    4797
$ITEMB  001114          1116#   4879*   4887    4898    4909    4957
$LF     001166          1141#   4644    4781    4790    4898
$LFLG   025477          4842*   4848#
$LPADR  001106          1113#
$LPERR  001110          1114#
$LSTCN= 177777           664#
$LSTIN= 000000           664#
$LSTST= 177777           664#
$LSTTA= 000000           664#
$MADR1  001222          1174#
$MADR2  001226          1178#
$MADR3  001232          1181#
$MADR4  001236          1184#
$MAIL   001170          1096    1100    1147#   1550    1563    4588    4885
$MAMS1  001220          1168#
$MAMS2  001224          1176#
$MAMS3  001230          1179#
$MAMS4  001234          1182#
$MBADR  001002          1096#
$MFLG   025476          4802*   4808    4843*   4847#
$MNEW   025222          2660    4669    4795#
$MSGAD  001204          1154#   4818*   4821
$MSGLG  001206          1155#   4823*
$MSGTY  001170          1148#   4816    4824*   4836    4840*
$MSWR   025211          4666    4793#
$MTYP1  001221          1169#
$MTYP2  001225          1177#
$MTYP3  001231          1180#
$MTYP4  001235          1183#
$NULL   001154          1134#   4615    4644
$OCNT   026530          5072*   5101*   5114#
$OMODE  026532          5067*   5071*   5076    5079*   5090*   5116#
$PASS   001176          1151#   1550*   2480*   2482
$PASTM  001006          1098#
$QUES   001164          1139#   2695    4644    4720    4774    4790    4898
$RDCHR  024714          4733#   5157
$RDDEC= ****** U        5159
$RDLIN  025034          4761#   5158
$RDOCT= ****** U        5159
```

```
$RDSZ = 000010          4754#
$R2A  = ****** U        5159
$SAVRE= ****** U        5159
$SETUP= 000106          1493#  1529  1530  1532  1534  1559  1560  4649  4797  4869  4895  4897
$STUP = 177777          1493#
$SWR  = 132000           664#   675  1138  1534  4861  4862  4863  4864  4865  4873  4880  4892  4896
                        4898
$SWREG  001212          1159#  1553
$TESTN  001174          1150#  4958*
$TKB    001146          1131#  2667  4647  4658  4675  4737  4743
$TKS    001144          1130#  1512* 2665  4647  4656  4672  4684* 4704* 4735  4741
$TN   = 000001           664#   675
$TPB    001152          1133#  4633* 4644
$TPFLG  001157          1137#  4582  4644
$TPS    001150          1132#  4631  4644
$TRAP   026534          1532   5125#
$TRAP2  026556          5136#  5147
$TRP  = 000013          5140#  5149#  5150#  5151#  5152#  5153#  5154  5155#  5156  5157#  5158#  5159#  5160#
$TRPAD  026570          5130   5147#
$TSTM   001004          1097#
$TSTNM  001102          1110#  4872  4898
$TTYIN  025162          4762   4763  4785  4789#
$TYPBN= ****** U        5153
$TYPDS  026062          4985#  5152
$TYPE   024110          4582#  4829  5140  5148
$TYPEC  024322          2679   4612  4619  4626  4631#  4632  4706
$TYPEX  024370          4637   4639  4642#
$TYPOC  026332          5070#  5149
$TYPON  026346          5069   5072#  5151
$TYPOS  026306          5065#  5150
$UNIT   001202          1153#
$UNITM  001010          1099#
$USWR   001214          1160#
$VECT1  001240          1185#
$VECT2  001242          1186#
$OFILL  026531          5066*  5070*  5080  5115#
$40CAT= ****** U        4882
.     = 026616          1055#  1059#  1063#  1066#  1069#  1071#  1073#  1078#  1084  1085#  1087#  1089#  1107#
                        1142   1455#  1456#  1473#  1474#  1527  2424  2471  2509#  2739  2765  2793  2806
                        2828   3883  3895  3911  3924  4027  4511#  4644  4647  4789#  4790  4797#  4850#
                        4898   4953#  5039#
.$ASTA= ****** U        4802   4805
.$X   = 001000          1084#  1089
```

```
COMMEN    798#
ENDCOM    798#
ERROR     692#  1666  1680  1685  1713  1718  1762  1800  1842  1848  1875  1910  1953  1987  2021
          2048  2056  2070  2084  2091  2096  2150  2169  2235  2260  2300  2305  2309  2322  2341
          2346  2358  2369  2374  2385  2411  2420  2425  2450  2455  2460  2868  2916  2968  3006
          3033  3059  3086  3132  3162  3238  3243  3252  3254  3315  3324  3328  3337  3339  3346
          3399  3406  3409  3414  3421  3612  3617  3626  3628  3689  3698  3702  3711  3713  3772
          3776  3781  3787  3893  3909  3922  3992  4011  4025
ESCAPE    798#
GETPRI    798#
GETSWR    798#  1560#
MULT      798#
NEWTST    798#
POP       798#  4844  4845  5026
PUSH      798#  4805  4807  4828  4985
REPORT    664#   798#
SCOPE     693#
SETPRI    798#
SETTRA   5140#  5149  5150  5151  5152  5154  5156  5157  5158  5159
SETUP     798#  1521
SKIP      798#
SLASH     798#
SPACE     798#
STARS     798#   802   804   825   827   867   869   921   923   954   956  1016  1018  1051  1053
          1081  1083  1090  1103  1142  1145  1441  1443  1459  1461  1643  1650  1748  1750  1770
          1772  1816  1818  1883  1885  1918  1920  1961  1963  1995  1997  2029  2036  2108  2118
          2193  2201  2278  2285  2325  2330  2390  2396  2427  2433  2476  2478  2614  2616  2653
          2655  2698  2705  2720  2724  2747  2751  2775  2779  2800  2802  2809  2813  2837  2839
          2853  2855  2876  2879  2901  2903  2924  2927  2951  2954  2961  2963  2995  2997  3022
          3024  3048  3050  3075  3077  3101  3103  3148  3155  3187  3190  3207  3209  3223  3229
          3279  3281  3293  3299  3355  3358  3370  3375  3436  3441  3477  3479  3506  3508  3517
          3519  3529  3531  3537  3539  3546  3549  3561  3564  3585  3587  3597  3603  3652  3654
          3667  3673  3723  3726  3737  3742  3797  3802  3839  3841  3865  3871  3955  3957  3979
          3982  3997  4000  4016  4018  4036  4038  4567  4646  4649  4725  4754  4800  4857  4900
          4975  5042  5119
SWRSU     798#  1534#
TRMTRP   5140#
TYPBIN    798#
TYPDEC    798#  2482  2485  2488  2491  2494  3458  3819  4940
TYPNAM    798#  1555
TYPNUM    798#
TYPOCS    798#
TYPOCT    798#  2658  4667  4912  4937
TYPTXT    798#
$$CMRE   1101#
$$CMTM   1101#
$$ESCA    798#
$$NEWT    798#
$$SET    5140#  5149  5150  5151  5152  5154  5156  5157  5158  5159
$$SETM   1550#
$$SKIP    798#
.EQUAT    664#   688
.HEADE    664#   665
.SETUP    664#  1493
```

```
        2619  2620  2631  2632  2658  2674  2707  2708  2709  2710  2711  2712  2713  2726  2727
        2729  2730  2754  2755  2782  2783  2804  2815  2816  2818  2819  2841  2846  2857  2859
        2862  2869  2881  2882  2884  2888  2898  2899  2905  2907  2910  2917  2929  2930  2932
        2936  2946  2947  2965  2971  2981  2983  2985  3000  3003  3027  3030  3053  3056  3080
        3083  3106  3127  3139  3159  3160  3161  3168  3169  3170  3192  3193  3196  3197  3198
        3211  3214  3215  3218  3220  3239  3240  3259  3266  3274  3275  3283  3286  3288  3289
        3308  3309  3317  3318  3360  3361  3362  3363  3364  3378  3379  3382  3385  3386  3447
        3458  3465  3510  3512  3513  3521  3522  3525  3526  3533  3534  3541  3542  3566  3567
        3577  3578  3579  3589  3592  3593  3594  3613  3614  3633  3640  3648  3649  3656  3659
        3661  3662  3682  3683  3691  3692  3728  3729  3730  3731  3732  3745  3746  3756  3759
        3760  3808  3819  3826  3880  3881  3942  3959  3972  3984  3987  3988  4002  4003  4004
        4005  4020  4021  4022  4586  4587  4592  4600  4615  4667  4684  4699  4704  4733  4734
        4761  4762  4770  4782  4783  4784  4785  4806  4807  4814  4818  4823  4824  4826  4828
        4838  4844  4845  4872  4877  4907  4912  4921  4926  4931  4932  4934  4937  4941  4948
        4949  4958  4986  4987  4988  4989  4990  4991  4992  4997  5000  5020  5026  5027  5028
        5029  5030  5032  5033  5065  5073  5074  5075  5081  5088  5106  5107  5108  5109  5110
        5125  5126  5130  5136  5137
MOVB    1569  2667  4597  4625  4633  4658  4675  4737  4743  4766  4776  4801  4802  4804  4879
        4887  4957  4995  4998  5012  5015  5024  5066  5067  5070  5071  5072  5076  5079  5080
        5099  5128
NEG     4994  5077
NOP     1864  2151  2170  2236  2261  2323  2359  2386
RESET   1612
ROL     2622  5083  5085  5086  5087  5089
RTI     1545  2677  2957  2969  2988  3001  3016  3028  3043  3054  3069  3081  3096  3115  3119
        3145  3163  3201  3221  3241  3276  3290  3310  3319  3367  3475  3493  3514  3523  3527
        3535  3543  3553  3557  3582  3595  3615  3650  3663  3684  3693  3735  3836  3855  4602
        4705  4753  4786  4897  5034  5111  5138
RTS     2596  2639  2651  2714  2742  2768  2796  2807  2831  2851  2870  2900  2918  2948  3976
        3994  4013  4031  4642  4846  4951  4965  5131
SUB     1658  2533  4821  4878  5001
SWAB    2885  2933  3194  3212  3284  3380  3568  3590  3657  3747  3965  3985
TRAP    5140  5149  5150  5151  5152  5154  5156  5157  5158  5159
TST     1561  1574  1577  1579  1629  1631  1638  1662  1663  1677  1697  1710  1776  1781  1803
        1810  1873  1922  2140  2159  2174  2176  2182  2184  2188  2273  2303  2344  2372  2423
        2458  2462  2500  2523  2524  2543  2544  2588  2672  2685  2756  2784  2860  2865  2891
        2908  2913  2939  2966  3110  3165  3231  3246  3301  3331  3395  3397  3412  3417  3571
        3573  3605  3620  3675  3705  3750  3752  3768  3770  3779  3783  3888  3904  3917  4599
        4607  4629  4697  4712  4816  4834  4836  4892  4943  5006  5016  5094  5127
TSTB    2665  3157  3882  4582  4631  4656  4672  4735  4741  4808  4819  4832  4935  5008  5022
.ASCII  1139  1140  4077  4086  4091  4097  4148  4170
.ASCIZ  1138  1141  1573  4040  4043  4045  4047  4049  4051  4053  4055  4058  4059  4060  4062
        4063  4064  4065  4066  4067  4068  4069  4074  4104  4111  4114  4116  4117  4118  4120
        4123  4125  4126  4129  4133  4138  4143  4151  4160  4176  4181  4183  4186  4190  4197
        4201  4205  4209  4213  4218  4223  4227  4230  4235  4240  4245  4250  4255  4260  4265
        4270  4275  4280  4282  4285  4288  4291  4294  4297  4300  4302  4304  4310  4316  4322
        4329  4336  4340  4347  4352  4358  4363  4366  4371  4377  4383  4389  4395  4401  4407
        4411  4415  4422  4425  4430  4435  4438  4441  4447  4450  4453  4460  4463  4467  4473
        4478  4483  4490  4500  4790  4791  4792  4793  4795  4952
.BLKB   4789
.BLKW   1455  1456  1473  1474  5039
.BYTE   1110  1111  1116  1117  1125  1126  1134  1135  1136  1137  1157  1158  1168  1169  1176
        1177  1179  1180  1182  1183  2508  4787  4788  4847  4848  4849  4889  4890  5112  5113
        5114  5115
```

ERRORS DETECTED: 0

```
*Z:CVKDAA/I,Z:CVKDAA.SEQ/CRF/SOL=CVKDAA.M11
RUN-TIME: 24 15 3 SECONDS
CORE USED:  25K
```