PDT-11

11 150 SYS EXER

CVKDAC0

AH F239C MC

FICHE 1 OF 1

MAY 1979

digital

MADE IN USA

```
        .REM %
```

IDENTIFICATION

PRODUCT CODE: AC-F238C-MC

PRODUCT NAME: CVKDACO PDT11/150 SYSTEM EXERCISER

PRODUCT DATE: APRIL 1979

MAINTAINER: DIAGNOSTIC ENGINEERING

C 1

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION.

VERSION "C" DIFFERS FROM PREVIOUS VERSION BECAUSE VERSION "C":
A. HAS ADDITIONAL FLOPPY ERROR CODES TO AID PRODUCTION.
B. SUPPORTS THE "EIS-FIS" INSTRUCTION SET OPTION.
C. ENHANCED MEMORY TESTING.

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 "EIS-FIS" LOGIC WILL BE EXERCISED IF SIZING DETERMINES IT TO BE PRESENT.

THE CONSOLE TERMINAL IS NOT TESTED.

AFTER THE MEMORY AND EIS-FIS (IF PRESENT) HAVE BEEN TESTED & ALL THE PERIPHERALS ARE
BEING EXERCISED & INTERRUPTING AT RANDOM, THE DISK SUBSYSTEM TESTS WILL BEGIN.
SEE TEST SUMMARY SEC. 7.0.

NOTE: IF RUNNING UNDER APT, THE SYNC COMM LOGIC WILL BE TESTED ONLY ON
THE 1'ST PASS. THIS IS SO APT 'BREAKS' SO NOT CAUSE FALSE SYNC COMM ERRORS.

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 (INCL. SCRIPT MODE)
WITH RT-11 MONITOR
WITH DIAGNOSTIC SUPERVISOR (NON SCRIPT MODE)

AN XXDP DRIVER IS NOT AVAILABLE FOR THE PDT-11/150
AT RELEASE TIME OF CVKDAC.
I. E. XXDP DOES NOT PRESENTLY SUPPORT THE PDT-11/150.

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 PROGRAM WILL AUTO-START AFTER BOOTING.

EXAMPLE OF STARTUP ASSUMING THAT:

```
        24K WORDS OF MEMORY
        EIS-FIS OPTION NOT PRESENT
        PRINTER NOT PRESENT
        CLUSTER TERM #2 NOT TO BE TESTED
        COMM PORT EXT. LOOPBACK
        CLUSTER TERM 1 & 3 INT. LOOPBACK.

        (STARTUP TYPEOUTS)

        CVKDAC  PDT-11/150 SYSTEM EXERCISER

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

        DEVM = 000017   NEW = 20005 <CR>        (CHANGE THE DEFAULTS  SEE SEC. 1.5)

        24K MEMORY PRESENT
        EIS-FIS OPTION NOT PRESENT
        TERM #2 TESTING DROPPED
        PRINTER NOT PRESENT

        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 FOLLOWED BY 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 ^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
     6 SET=        100= DROP EIS-FIS 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 15-6 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.
5 MIN TO WRITE, READ & DATA COMPARE ON TRACKS 0-4, 40-44, 72-76(10) ON BOTH DRIVES.
5 - 60 SEC TO DO 10 RANDOM SEEKS ON EACH DISK BETWEEN 1'ST 5 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 PING 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.

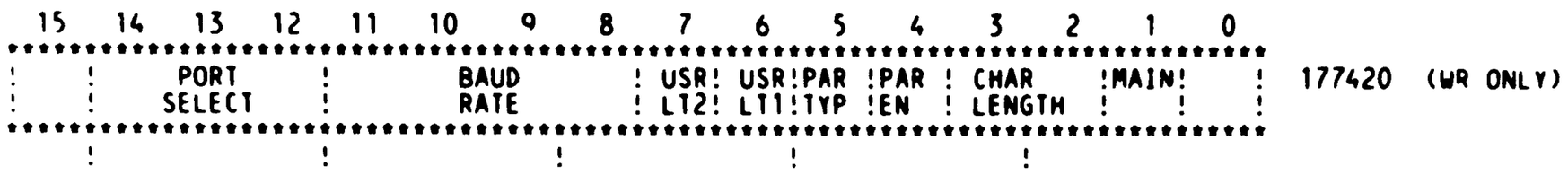
4.0 PROGRESS & PERFORMANCE REPORTS

THE FOLLOWING REPORTS ARE ENABLED WITH 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):

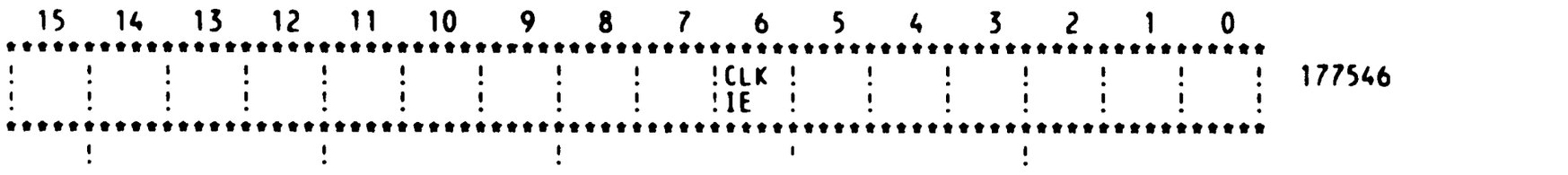
MEM TESTS DONE	(ALL MEMORY FROM LOCATION 0 TO THE BEGINNING OF THE RT-11/XXDP MONITOR (WHEN AVAIL) HAS BEEN TESTED.)
EIS-FIS TESTS DONE	(1000. NUMBER OF PASSES OF THE EIS-FIS INSTRUCTION EXERCISER HAS BEEN EXECUTED.)
CLOCK RUNNING	(100 INTERRUPTS HAVE BEEN DETECTED BEFORE EXERCISING THE NEXT DEVICE. 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) (FOR 1'ST PASS TRACKS 0-4, 40-44, 72-76(10) DONE ONLY.)
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.

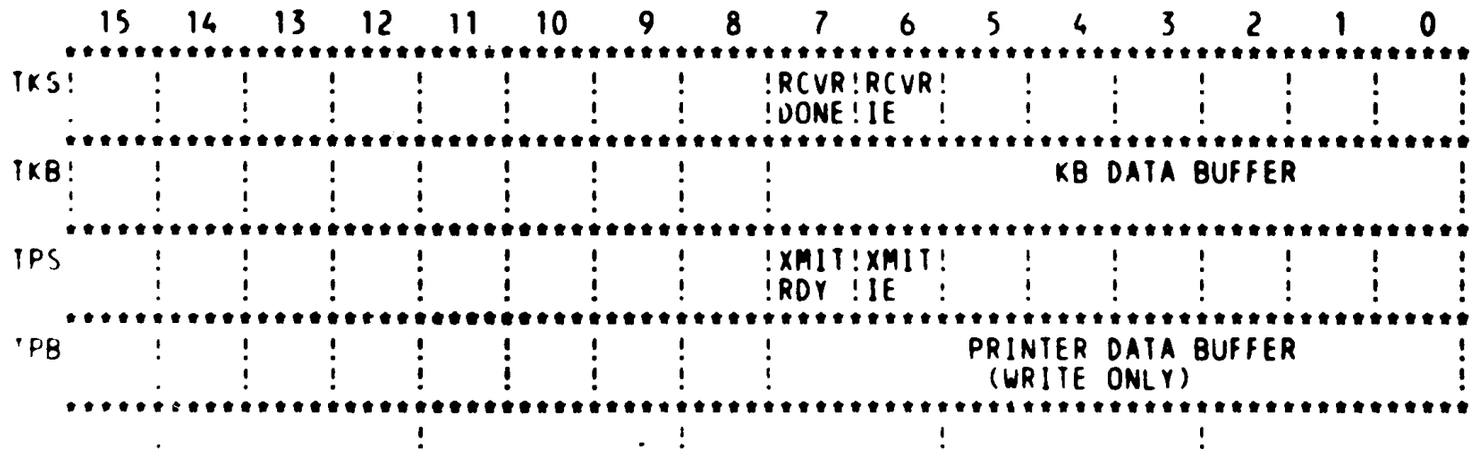
PARAMETER REGISTER:



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)
FCSR		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

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 125252 & 052525 PATTERN.
C. IF 28K PRESENT, MEMORY FROM 28K TO 30K IS SIMILARLY TESTED.

PHASE 2

EIS-FIS TESTS
A. EIS INSTRUCTIONS "ASH-ASHC-MUL-DIV" ARE EXECUTED.
B. FIS INSTRUCTIONS "FADD-FSUB-FMUL-FDIV" ARE EXECUTED.

PHASE 3

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 4

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  CVKDAC          PDT11-150 EXERCISER
                        ;*COPYRIGHT (C) 1979
                        ;*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
```

```
                                                    D 2
747     000000          PR0=    0               ;;PRIORITY LEVEL 0
748     000040          PR1=    40              ;;PRIORITY LEVEL 1
749     000100          PR2=    100             ;;PRIORITY LEVEL 2
750     000140          PR3=    140             ;;PRIORITY LEVEL 3
751     000200          PR4=    200             ;;PRIORITY LEVEL 4
752     000240          PR5=    240             ;;PRIORITY LEVEL 5
753     000300          PR6=    300             ;;PRIORITY LEVEL 6
754     000340          PR7=    340             ;;PRIORITY LEVEL 7
755
756                     ;*"SWITCH REGISTER" SWITCH DEFINITIONS
757     100000          SW15=   100000
758     040000          SW14=   40000
759     020000          SW13=   20000
760     010000          SW12=   10000
761     004000          SW11=   4000
762     002000          SW10=   2000
763     001000          SW09=   1000
764     000400          SW08=   400
765     000200          SW07=   200
766     000100          SW06=   100
767     000040          SW05=   40
768     000020          SW04=   20
769     000010          SW03=   10
770     000004          SW02=   4
771     000002          SW01=   2
772     000001          SW00=   1
773                     .EQUIV  SW09,SW9
774                     .EQUIV  SW08,SW8
775                     .EQUIV  SW07,SW7
776                     .EQUIV  SW06,SW6
777                     .EQUIV  SW05,SW5
778                     .EQUIV  SW04,SW4
779                     .EQUIV  SW03,SW3
780                     .EQUIV  SW02,SW2
781                     .EQUIV  SW01,SW1
782                     .EQUIV  SW00,SW0
783
784                     ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
785     100000          BIT15=  100000
786     040000          BIT14=  40000
787     020000          BIT13=  20000
788     010000          BIT12=  10000
789     004000          BIT11=  4000
790     002000          BIT10=  2000
791     001000          BIT09=  1000
792     000400          BIT08=  400
793     000200          BIT07=  200
794     000100          BIT06=  100
795     000040          BIT05=  40
796     000020          BIT04=  20
797     000010          BIT03=  10
798     000004          BIT02=  4
799     000002          BIT01=  2
800     000001          BIT00=  1
801                     .EQUIV  BIT09,BIT9
802                     .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 FQUATES      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
000200          RDY=    BIT7            ;XMIT RDY
000100          IE=     BIT6            ;XMIT IE
000040          DNAIE=  BIT5            ;DATA NOT AVAIL IE
000020          SEND=   BIT4            ;SEND
000010          HFDUP=  BIT3            ;1=HALF DUPLEX, 0=FULL DUPLEX (DEFAULT = 0 FOR TESTS)



                                ;;*****************************************************************
                                ;DISK BIT DEFINITIONS
                                ;;*****************************************************************

                                ;RXCS
                                ;
100000          ERR=    BIT15           ;ERROR
000200          DONE=   BIT7            ;CONTR RDY
000100          IE=     BIT6            ;DRIVE IE
000020          DX1=    BIT4            ;UNIT SELECT: 0=DX0, 1=DX1
000020          USEL=   BIT4            ;UNIT SELECT

                                ;FUNCTIONS (INCLUDES BIT0 = GO)

000001          FBUF=   1               ;FILL BUFFER
000003          EBUF=   3               ;EMPTY BUFFER
000005          WSEC=   5               ;WRITE SECTOR
000007          RSEC=   7               ;READ SECTOR
000011          INITAL= 11              ;INITIALIZE
000013          RSTAT=  13              ;READ STATUS
000015          WDDSEC= 15              ;WRITE DELETED DATA SECTOR
000017          RESTOR= 17              ;RESTORE TO TRACK 0

                                ;RXES: ERROR & STATUS
                                ;
000200          RDY=    BIT7            ;DRIVE RDY
000040          DD=     BIT5            ;DELETED DATA DET.
000020          RNF=    BIT4            ;RECORD NOT FOUND
000010          CRC=    BIT3            ;CRC ERROR
000004          LDATA=  BIT2            ;LOST DATA
000002          INVADR= BIT1            ;INVALID ADDR
000001          ID=     BIT0            ;INITIALIZE DONE
```

```
                                ;;*****************************************************************
                                ;*      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  003334                  JMP     START   ;USE ONLY ONCE. WILL BE OVERLAID BY
                                                        ;PRINTER VECTOR ADDR.
                        000240          .=240
000240  000137  003334                  JMP     START   ;RESTART ADDR

                        000250          .=250
000250  000137  003316                  JMP     START3  ;ENTER HERE FOR COPY UTILITY
                        000260          .=260
000260  000137  003276                  JMP     START1  ;COMPATABILTIY PASS 1
                        000270          .=270
000270  000137  003304                  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)
```

```
 1126                                   .SBTTL  COMMON TAGS
 1127
 1128                                   ;;**********************************************************************
 1129                                   ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
 1130                                   ;*USED IN THE PROGRAM.
 1131
 1132            001100                         .=1100
 1133  001100                           $CMTAG:                         ;;START OF COMMON TAGS
 1134  001100  000000                          .WORD   0
 1135  001102     000                   $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
 1136  001103     000                   $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
 1137  001104  000000                   $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
 1138  001106  000000                   $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
 1139  001110  000000                   $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
 1140  001112  000000                   $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
 1141  001114     000                   $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
 1142  001115     001                   $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
 1143  001116  000000                   $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
 1144  001120  000000                   $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
 1145  001122  000000                   $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
 1146  001124  000000                   $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
 1147  001126  000000                   $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
 1148  001130  000000                           .WORD   0               ;;RESERVED--NOT TO BE USED
 1149  001132  000000                           .WORD   0
 1150  001134     000                   $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
 1151  001135     000                   $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
 1152  001136  000000                           .WORD   0
 1153  001140  177570                   SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
 1154  001142  177570                   DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
 1155  001144  177560                   $TKS:   177560                  ;;TTY KBD STATUS
 1156  001146  177562                   $TKB:   177562                  ;;TTY KBD BUFFER
 1157  001150  177564                   $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
 1158  001152  177566                   $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
 1159  001154     000                   $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
 1160  001155     002                   $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
 1161  001156     012                   $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
 1162  001157     000                   $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
 1163  001160  177607  000377           $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
 1164  001164     077                   $QUES:  .ASCII  /?/             ;;QUESTION MARK
 1165  001165     015                   $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
 1166  001166  000012                   $LF:    .ASCIZ  <12>            ;;LINE FEED
 1167                                   ;;**********************************************************************
 1168                                   .SBTTL  APT MAILBOX-ETABLE
 1169
 1170                                   ;;**********************************************************************
 1171                                   .EVEN
 1172  001170                           $MAIL:                          ;;APT MAILBOX
 1173  001170  000000                   $MSGTY: .WORD   AMSGTY           ;;MESSAGE TYPE CODE
 1174  001172  000000                   $FATAL: .WORD   AFATAL           ;;FATAL ERROR NUMBER
 1175  001174  000000                   $TESTN: .WORD   ATESTN           ;;TEST NUMBER
 1176  001176  000000                   $PASS:  .WORD   APASS            ;;PASS COUNT
 1177  001200  000000                   $DEVCT: .WORD   ADEVCT           ;;DEVICE COUNT
 1178  001202  000000                   $UNIT:  .WORD   AUNIT            ;;I/O UNIT NUMBER
 1179  001204  000000                   $MSGAD: .WORD   AMSGAD           ;;MESSAGE ADDRESS
 1180  001206  000000                   $MSGLG: .WORD   AMSGLG           ;;MESSAGE LENGTH
 1181  001210                           $ETABLE:                        ;;APT ENVIRONMENT TABLE
```

```
1182  001210  000                 $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1183  001211  000                 $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1184  001212  000000              $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1185  001214  000000              $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1186  001216  000000              $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1187                              ;*                      BITS 15-11=CPU TYPE
1188                              ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1189                              ;*                              11/70=06,PDQ=07,Q=10
1190                              ;*                      BIT 10=REAL TIME CLOCK
1191                              ;*                      BIT  9=FLOATING POINT PROCESSOR
1192                              ;*                      BIT  8=MEMORY MANAGEMENT
1193  001220  000                 $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1194  001221  000                 $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1195                              ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1196                              ;*                              900 NSEC CORE=001
1197                              ;*                              300 NSEC BIPOLAR=002
1198                              ;*                              500 NSEC MOS=003
1199  001222  000000              $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1200                              ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
1201  001224  000                 $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1202  001225  000                 $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1203  001226  000000              $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1204  001230  000                 $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1205  001231  000                 $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1206  001232  000000              $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1207  001234  000                 $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1208  001235  000                 $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1209  001236  000000              $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1210  001240  000300              $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1211  001242  000000              $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1212  001244  176500              $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1213  001246  000017              $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1214  001250                      $ETEND:
1215                              .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  021511  024207  024614                  EM1,DH2,DT2,DF
001256  025110      \
                                        ;ERR 2
001260  021511  024207  024614                  EM1,DH2,DT2,DF
001266  025110
                                        ;ERR 3
001270  021530  024233  024624                  EM2,DH3,DT3,DF
001276  025110
                                        ;ERR 4
001300  021511  024207  024614                  EM1,DH2,DT2,DF
001306  025110
                                        ;ERR 5
001310  021530  024233  024624                  EM2,DH3,DT3,DF
001316  025110
                                        ;ERR 6
001320  021552  024170  024606                  EM3,DH1,DT1,DF
001326  02511
                                        ;ERR 7
001330  021   6   024275  024640                EM4,DH4,DT4,DF
001336  025 10
                                        ;ERR 10
001340  021641  024326  024650                  EM5,DH5,DT5,DF
001346  025110
                                        ;ERR 11
001350  021667  024326  024662                  EM6,DH5,DT6,DF
001356  025110
                                        ;ERR 12
001360  021715  024326  024674                  EM7,DH5,DT7,DF
001366  025110
                                        ;ERR 13
001370  021743  024326  024706                  EM8,DH5,DT8,DF
001376  025110
1   5                                   ;ERR 14
001400  021774  024326  024706                  EM9,DH5,DT8,DF
001406  025110
                                        ;ERR 15
001410  022024  024363  024720                  EM10,DH6,DT9,DF
001416  025110
```

```
1271                                        ;ERR 16
1272  001420  022051  024430  024736              EM11,DH7,DT10,DF1
1273  001426  025116
1274                                        ;ERR 17
1275  001430  022051  024430  024736              EM11,DH7,DT10,DF1
1276  001436  025116
1277                                        ;ERR 20
1278  001440  022067  024430  024736              EM13,DH7,DT10,DF1
1279  001446  025116
1280                                        ;ERR 21
1281  001450  022120  024430  024736              EM14,DH7,DT10,DF1
1282  001456  025116
1283                                        ;ERR 22
1284  001460  022151  024517  024760              EM15,DH8,DT11,DF2
1285  001466  025126
1286                                        ;ERR 23
1287  001470  022202  024517  024760              EM16,DH8,DT11,DF2
1288  001476  025126
1289                                        ;ERR 24
1290  001500  022233  024430  025002              EM17,DH7,DT12,DF1
1291  001506  025116
1292                                        ;ERR 25
1293  001510  022233  024430  025002              EM17,DH7,DT12,DF1
1294  001516  025116
1295                                        ;ERR 26
1296  001520  022251  024430  025002              EM19,DH7,DT12,DF1
1297  001526  025116
1298                                        ;ERR 27
1299  001530  022303  024430  025002              EM20,DH7,DT12,DF1
1300  001536  025116
1301                                        ;ERR 30
1302  001540  022333  024517  025024              EM21,DH8,DT13,DF2
1303  001546  025126
1304                                        ;ERR 31
1305  001550  022364  024517  025024              EM22,DH8,DT13,DF2
1306  001556  025126
1307                                        ;ERR 32
1308  001560  022415  024170  024606              EM23,DH1,DT1,DF
1309  001566  025110
1310                                        ;ERR 33
1311  001570  022426  024275  024640              EM24,DH4,DT4,DF
1312  001576  025110
1313                                        ;ERR 34
1314  001600  022443  024170  024606              EM25,DH1,DT1,DF
1315  001606  025110
1316                                        ;ERR 35
1317  001610  022460  024170  024606              EM26,DH1,DT1,DF
1318  001616  025110
1319                                        ;ERR 36
1320  001620  022475  024170  024606              EM27,DH1,DT1,DF
1321  001626  025110
```

```
1322                                        ;ERR 37
1323  001630  022512  024170  024606              EM28,DH1,DT1,DF
1324  001636  025110
1325                                        ;ERR 40
1326  001640  022531  024170  024606              EM29,DH1,DT1,DF
1327  001646  025110
1328                                        ;ERR 41
1329  001650  022551  024363  025046              EM30,DH6,DT14,DF
1330  001656  025110
1331                                        ;ERR 42
1332  001660  022562  024363  025046              EM31,DH6,DT14,DF
1333  001666  025110
1334                                        ;ERR 43
1335  001670  022573  024170  024606              EM32,DH1,DT1,DF
1336  001676  025110
1337                                        ;ERR 44
1338  001700  022636  024170  024606              EM33,DH1,DT1,DF
1339  001706  025110
1340                                        ;ERR 45
1341  001710  022677  024170  024606              EM34,DH1,DT1,DF
1342  001716  025110
1343                                        ;ERR 46
1344  001720  022551  024363  025046              EM30,DH6,DT14,DF
1345  001726  025110
1346                                        ;ERR 47
1347  001730  022551  024363  025046              EM30,DH6,DT14,DF
1348  001736  025110
1349                                        ;ERR 50
1350  001740  022551  024363  025046              EM30,DH6,DT14,DF
1351  001746  025110
1352                                        ;ERR 51
1353  001750  022551  024363  025046              EM30,DH6,DT14,DF
1354  001756  025110
1355                                        ;ERR 52
1356  001760  022736  024363  024720              EM35,DH6,DT9,DF
1357  001766  025110
1358                                        ;ERR 53
1359  001770  023003  024517  024760              EM36,DH8,DT11,DF2
1360  001776  025126
1361                                        ;ERR 54
1362  002000  023052  024363  024720              EM37,DH6,DT9,DF
1363  002006  025110
1364                                        ;ERR 55
1365  002010  023076  024363  024720              EM38,DH6,DT9,DF
1366  002016  025110
1367                                        ;ERR 56
1368  002020  022551  024363  025046              EM30,DH6,DT14,DF
1369  002026  025110
1370                                        ;ERR 57
1371  002030  023144  024363  024720              EM39,DH6,DT9,DF
1372  002036  025110
```

```
1373                                        ;ERR 60
1374  002040  022551  024363  025046              EM30,DH6,DT14,DF
1375  002046  025110
1376                                        ;ERR 61
1377  002050  023174  024363  024720              EM40,DH6,DT9,DF
1378  002056  025110
1379                                        ;ERR 62
1380  002060  023237  024363  024720              EM41,DH6,DT9,DF
1381  002066  025110
1382                                        ;ERR 63
1383  002070  023273  024363  024720              EM42,DH6,DT9,DF
1384  002076  025110
1385                                        ;ERR 64
1386  002100  023314  024430  024736              EM43,DH7,DT10,DF1
1387  002106  025116
1388                                        ;ERR 65
1389  002110  023314  024430  024736              EM43,DH7,DT10,DF1
1390  002116  025116
1391                                        ;ERR 66
1392  002120  023346  024430  024736              EM45,DH7,DT10,DF1
1393  002126  025116
1394                                        ;ERR 67
1395  002130  023346  024430  024736              EM45,DH7,DT10,DF1
1396  002136  025116
1397                                        ;ERR 70
1398  002140  023377  024430  025002              EM47,DH7,DT12,DF1
1399  002146  025116
1400                                        ;ERR 71
1401  002150  023377  024430  025002              EM47,DH7,DT12,DF1
1402  002156  025116
1403                                        ;ERR 72
1404  002160  023431  024430  025002              EM49,DH7,DT12,DF1
1405  002166  025116
1406                                        ;ERR 73
1407  002170  023431  024430  025002              EM49,DH7,DT12,DF1
1408  002176  025116
1409                                        ;ERR 74
1410  002200  023462  024363  024720              EM51,DH6,DT9,DF
1411  002206  025110
1412                                        ;ERR 75
1413  002210  023506  024363  025046              EM52,DH6,DT14,DF
1414  002216  025110
1415                                        ;ERR 76
1416  002220  023532  024326  025064              EM53,DH5,DT15,DF
1417  002226  025110
1418                                        ;ERR 77
1419  002230  023532  024326  025076              EM53,DH5,DT16,DF
1420  002236  025110
```

```
1421
1422                                    ; ERRORS FOR COPY UTILITY
1423
1424                                    ;ERR 100
1425  002240  022067  024363  025046            EM13,DH6,DT14,DF
1426  002246  025110
1427                                    ;ERR 101
1428  002250  022233  024363  025046            EM17,DH6,DT14,DF
1429  002256  025110
1430                                    ;ERR 102
1431  002260  023601  024363  025046            EM54,DH6,DT14,DF
1432  002266  025110
1433                                    ;ERR 103
1434  002270  022251  024363  025046            EM19,DH6,DT14,DF
1435  002276  025110
1436
1437                                    ;BAD SECTOR/TRACK TABLE FULL ERRORS
1438
1439                                    ;ERR 104
1440  002300  023616  000000  000000            EM55,0,0,0
1441  002306  000000
1442                                    ;ERR 105
1443  002310  023653  000000  000000            EM56,0,0,0
1444  002316  000000
1445
1446                                    ;CRC ERRORS
1447
1448                                    ;ERR 106
1449  002320  023710  024430  024736            EM57,DH7,DT10,DF1
1450  002326  025116
1451                                    ;ERR 107
1452  002330  023731  024517  024760            EM58,DH8,DT11,DF2
1453  002336  025126
1454                                    ;ERR 110
1455  002340  023752  000000  000000            EM59,0,0,0
1456  002346  000000
1457                                    ;ERR 111
1458  002350  024010  024430  025002            EM60,DH7,DT12,DF1
1459  002356  025116
1460                                    ;ERR 112
1461  002360  024031  024517  025024            EM61,DH8,DT13,DF2
1462  002366  025126
1463                                    ;ERR 113
1464  002370  024052  000000  000000            EM62,0,0,0
1465  002376  000000
1466                                    ;ERR 114
1467  002400  023506  024363  025046            EM52,DH6,DT14,DF
1468  002406  025110
1469                                    ;ERR 115
1470  002410  023506  024363  025046            EM52,DH6,DT14,DF
1471  002416  025110
1472                                    ;ERR 116
1473  002420  023506  024363  025046            EM52,DH6,DT14,DF
1474  002426  025110
1475                                    ;ERR 117
1476  002430  022551  024363  025046            EM30,DH6,DT14,DF
```

```
1477  002436  025110
1478                                  ;ERR 120
1479  002440  022562  024363  025046          EM31,DH6,DT14,DF
1480  002446  025110
1481                                  ;ERR 121
1482  002450  022551  024363  025046          EM30,DH6,DT14,DF
1483  002456  025110
1484                                  ;ERR 122
1485  002460  022562  024363  025046          EM31,DH6,DT14,DF
1486  002466  025110
1487                                  ;ERR 123
1488  002470  022562  024363  025046          EM31,DH6,DT14,DF
1489  002476  025110
1490                                  ;ERR 124
1491  002500  023506  024363  025046          EM52,DH6,DT14,DF
1492  002506  025110
1493                                  ;ERR 125
1494  002510  023506  024363  025046          EM52,DH6,DT14,DF
1495  002516  025110
1496
1497
1498                                  ;EIS FIS ERRORS
1499
1500
1501                                  ;ERR 126
1502  002520  024110  024170  024606          EM63,DH1,DT1,DF
1503  002526  025110
1504                                  ;ERR 127
1505  002530  024145  024170  024606          EM64,DH1,DT1,DF
1506  002536  025110
1507
```

```
1508                            ;;*****************************************************************
1509                            ;REGISTERS & BUFFERS FOR DX0 TESTS
1510                            ;;*****************************************************************
1511
1512  002540  000000            D0PAT:  0               ;DX0 PATT TEST DONE FLAG
1513  002542  000000            D0CNT:  0               ;DX0 RANDOM SEEK COUNTER
1514  002544  000000            D0ERR:  0               ;DX0 RCSR ERR COUNTER
1515  002546  000000            D0TRK:  0               ;DX0 TRACK
1516  002550  000000            D0SEC:  0               ;DX0 SECTOR
1517  002552  000000            D0CMER: 0               ;DX0 DATA COMPARE ERROR FLAG
1518  002554  000000            D0CERR: 0               ;DX0 TOTAL DATA COMP ERROR CT
1519  002556  000000            D0EXP:  0               ;DX0 EXPECTED DATA
1520  002560  000000            D0REC:  0               ;DX0 RECEIVED DATA
1521  002562  000000            D0CRC:  0               ;DX0 CRC ERROR FLAG
1522  002564  000100            D0BUF:  .BLKW 100       ;DX0 DATA BUFFER
1523  002764  000012            D0BAD:  .BLKW 10.       ;DX0 BAD TRK/SEC TABLE
1524
1525
1526                            ;;*****************************************************************
1527                            ;REGISTERS & BUFFERS FOR DX1 TESTS
1528                            ;;*****************************************************************
1529
1530  003010  000000            D1PAT:  0               ;DX1 PATT TEST DONE FLAG
1531  003012  000000            D1CNT:  0               ;DX1 RANDOM SEEK COUNTER
1532  003014  000000            D1ERR:  0               ;DX1 RCSR ERR COUNTER
1533  003016  000000            D1TRK:  0               ;DX1 TRACK
1534  003020  000000            D1SEC:  0               ;DX1 SECTOR
1535  003022  000000            D1CMER: 0               ;DX1 DATA COMPARE ERROR FLAG
1536  003024  000000            D1CERR: 0               ;DX1 TOTAL DATA COMPARE ERROR CT
1537  003026  000000            D1EXP:  0               ;DX1 EXPECTED DATA
1538  003030  000000            D1REC:  0               ;DX1 RECEIVED DATA
1539  003032  000000            D1CRC:  0               ;DX1 CRC ERROR FLAG
1540  003034  000100            D1BUF:  .BLKW 100       ;DX1 DATA BUFFER
1541  003234  000012            D1BAD:  .BLKW 10.       ;DX1 BAD TRK/SEC TABLE
1542
1543  003260  000000            DXTST:  0               ;1 = DATA PATT TEST
1544                                                    ;2 = RANDOM SEEK TEST
1545                                                    ;3 = INITIALIZE TEST
1546                                                    ;4 = RESTORE TEST
1547                                                    ;5 = WRITE DELETED DATA TEST
1548                                                    ;6 = INVALID ADDRESS TEST
1549
1550  003262  000000            FIN:    0               ;COMMON FLG FOR DX0 & DX1 FOR TRK/SEC FINISHED
1551  003264  000000            MAXTRK: 0               ;MAX TRACK #
1552                                                    ;0 FOR 1'ST PASS, 114 FOR SUBSEQUENT PASSES. SET AT START.
1553  003266  000000            MAXSK:  0               ;# OF RANDOM SEEKS TO BE PERFORMED.
1554                                                    ;10 FOR 1'ST PASS, 500 FOR SUBSEQUENT PASSES. SET AT EOP.
1555  003270  000000            COPFLG: 0               ; 0 = NORM TEST   1 = COPY UTILITY ACTIVE
1556  003272  000000            COMP1:  0               ;1 = COMPAT TESTS, PASS 1
1557  003274  000000            COMP2:  0               ;1                 PASS 2
1558
1559
```

```
                                        .SBTTL  PROGRAM

003276  005237  003272        START1: INC     COMP1           ;COMPAT PASS 1
003302  000424                        BR      S1

003304  005237  003274        START2: INC     COMP2           ;COMPAT PASS 2
003310  005037  003272                CLR     COMP1
003314  000421                        BR      S2

003316  005237  003270        START3: INC     COPFLG          ;COPY UTILITY
003322  005037  003272                CLR     COMP1
003326  005037  003274                CLR     COMP2
003332  000414                        BR      S3

                                ;NORMAL START AND RESTART
003334  012737  000102  000100 START:  MOV     #CLKVEC+2,CLKVEC        ;DO RTI IF CLOCK INTERR
003342  012737  000002  000102        MOV     #RTI,CLKVEC+2

003350  005037  003272                CLR     COMP1           ;NORMAL TESTING
003354  005037  003274        S1:     CLR     COMP2
003360  005037  003270        S2:     CLR     COPFLG

003364  012706  001100        S3:     MOV     #STACK,SP       ;SETUP STACK POINTER
003370  042777  000100  175546        BIC     #100,@$TKS      ;DISABLE ANY TTY INTERRUPTS
                                                              ;TO RT-11 MONITOR
                                        ;MTPS   #PR4            ;DISABLE INTERRUPTS
003376  106427                        .WORD   106427
003400  000200                                PR4
003402  013737  000042  004550        MOV     42,HLD42        ;SAVE
003410  123727  001210  000001        CMPB    $ENV,#1         ;APT?
003416  001402                        BEQ     2$              ;BR IF YES
003420  005037  000042                CLR     42              ;ELSE CLR SO WE CAN USE SOFTSWR
003424                          2$:
                                .SBTTL  INITIALIZE THE COMMON TAGS
                                ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003424  012706  001100                MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003430  005026                        CLR     (R6)+           ;;CLEAR MEMORY LOCATION
003432  022706  001140                CMP     #SWR,R6 ;;DONE?
003436  001374                        BNE     .-6             ;;LOOP BACK IF NO
003440  012706  001100                MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                ;;INITIALIZE A FEW VECTORS
003444  012737  026530  000030        MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003452  012737  000340  000032        MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003460  012737  027562  000034        MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003466  012737  000340  000036        MOV     #340,@#TRAPVEC+2;LEVEL 7
                                ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
003474  013746  000004                MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003500  012737  003534  000004        MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003506  012737  177570  001140        MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003514  012737  177570  001142        MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003522  022777  177777  175410        CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003530  001012                        BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                              ;;AND  THE HARDWARE SWR IS NOT = -1
003532  000403                        BR      65$             ;;BRANCH IF NO TIMEOUT
003534  012716  003542        64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
```

```
1616  003540  000002                          RTI
1617  003542  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1618  003550  012737  000174  001142          MOV     #DISPREG,DISPLAY
1619  003556  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1620
1621  003562  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
1622  003566  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1623  003574  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1624  003576  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1625  003604                          67$:
1626                                  .SBTTL  TYPE PROGRAM NAME
1627                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1628  003604  005227  177777                  INC     #-1             ;;FIRST TIME?
1629  003610  001046                          BNE     68$             ;;BRANCH IF NO
1630  003612  104401  003660                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1631                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1632  003616  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1633  003622  001012                          BNE     70$             ;;BRANCH IF YES
1634  003624  123727  001210  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1635  003632  001406                          BEQ     70$             ;;BRANCH IF YES
1636  003634  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1637  003642  001005                          BNE     71$             ;;BRANCH IF NO
1638  003644  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1639  003646  000403                          BR      71$
1640  003650  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1641  003656                          71$:
1642  003656  000423                          BR      68$             ;;GET OVER THE ASCIZ
1643                                  ;;69$:  .ASCIZ  <CRLF>*CVKDAC  PDT-11/150 SYSTEM EXERCISER*<CRLF>
1644  003726                          68$:
1645  003726  005737  003270                  TST     COPFLG          ;COPY UTILITY?
1646  003732  001057                          BNE     LOOP            ;BR IF YES
1647
1648  003734  005737  003272                  TST     COMP1           ;COMPAT PASS 1?
1649  003740  001054                          BNE     LOOP            ;BR IF YES
1650  003742  005737  003274                  TST     COMP2           ;COMPAT PASS 2?
1651  003746  001051                          BNE     LOOP            ;BR IF YES
1652
1653  003750  012737  000004  003264          MOV     #4,MAXTRK       ;TRK 0-4, 40-44, 72-76(10) FOR 1'ST PASS
1654  003756  012737  000012  003266          MOV     #10.,MAXSK      ;10 RANDOM SEEKS FOR 1'ST PASS
1655
1656  003764  005037  027100                  CLR     TERR            ;TOTAL ERR CT
1657  003770  005037  027102                  CLR     TSERR           ;TOTAL SOFT ERR CT FOR EOP TYPEOUT
1658
1659  003774  052737  000006  176610          BIS     #<RTS!DTR>,AMRC
1660  004002  042737  000006  176610          BIC     #<RTS!DTR>,AMRC ;DTR MUST BE TOGGLED.
```

```
1661
1662  004010  005227  177777          INC     #-1             ;1'ST TIME?
1663  004014  001005                  BNE     4$              ;BR IF NO
1664  004016  123727  001210  000001  CMPB    $ENV,#1         ;APT?
1665  004024  001401                  BEQ     4$              ;BR IF YES
1666  004026  104412                  GTDEVM                  ;SHOW CURRENT DEVM & GET NEW
1667
1668  004030  004737  011340  4$:     JSR     PC,SIZE         ;SEE WHO IS ON THE SYSTEM
1669  004034  123727  001210  000001  CMPB    $ENV,#1         ;ARE WE RUNNING UNDER APT?
1670  004042  001413                  BEQ     LOOP            ;BR IF YES & DONT HALT
1671  004044  032737  001000  001246  BIT     #1000,$DEVM     ;DX0 THERE?
1672  004052  001404                  BEQ     1$              ;BR IF YES
1673  004054  032737  000400  001246  BIT     #400,$DEVM      ;DX1 THERE?
1674  004062  001003                  BNE     LOOP            ;BR IF NO
1675  004064  104401  020365  1$:     TYPE    ,WARNING        ;INSERT SCRATCH DISKS
1676  004070  000000                  HALT
1677
1678
1679  004072  012706  001100  LOOP:   MOV     #1100,SP        ;LOOP HERE AFTER EOP & RESET STACK
1680
1681  004076  004737  012622          JSR     PC,TIMER1       ;TIMER TO ALLOW 'P' TO PRINT BEFORE RESET
1682
1683  004102  000005                  RESET                   ;CLEAR ALL INTERRUPT ENABLES
1684  004104  042737  000100  177546  BIC     #IE,CLK         ;RESET SETS CLK IE. DONT ALLOW YET
1685
1686                                  ;MTPS   #PR0            ;ALLOW INTERRUPTS
1687  004112  106427                  .WORD   106427
1688  004114  000000                  PR0
1689
1690  004116  052737  000006  176610  BIS     #<RTS!DTR>,AMRC ;TO CONDITION ALL TERMINALS &
1691                                                          ;COMM PORT TO OPERATE IN EXT LOOPBACK MODE.
1692                                                          ;NO HARM IF LOOPBACK NOT USED.
1693  004124  004737  012622          JSR     PC,TIMER1       ;TIMER TO ALLOW PREV XFERS TO FINISH
1694
1695  004130  005037  027104          CLR     PERR            ;PASS ERR COUNT FOR EOP TYPEOUT
1696  004134  005037  027106          CLR     PSERR           ;PASS SOFT ERR COUNT FOR EOP TYPEOUT
1697
1698  004140  004737  012720          JSR     PC,CLRBAD       ;CLEAR BAD SECTOR/TRACK TABLES
1699
1700  004144  005737  003272          TST     COMP1           ;COMPAT PASS 1?
1701  004150  001003                  BNE     1$              ;BR IF YES
1702  004152  005737  003274          TST     COMP2           ;COMPAT PASS 2?
1703  004156  001405                  BEQ     2$              ;BR IF NO
1704
1705  004160  012737  000114  003264  1$:     MOV     #114,MAXTRK
1706  004166  000137  007246          JMP     DXTST1          ;COMPATABILITY TESTING
1707
1708  004172  005737  003270  2$:     TST     COPFLG          ;COPY UTILITY?
1709  004176  001402                  BEQ     MEMTST          ;BR IF NO...NORMAL TESTING
1710  004200  000137  017410          JMP     COPY            ;ELSE GO COPY
1711
```

```
                                        .SBTTL  PROGRAM
                                        ;;*****************************************************************
                                        ;       MEMORY TESTS
                                        ;
                                        ; 1. MEMORY FROM 0 TO THE LAST LOC OF PROGRAM (LSTAD) IS TESTED FOR TIMEOUT.
                                        ; 2. MEMORY FROM 'LSTAD' TO THE BOTTOM OF THE RT-11/XXDP MONITOR
                                        ;    IS TESTED BY A PASS OF A 125252 PATT FOLLOWED BY A PASS OF 052525 PATTERN.
                                        ; 3. MEMORY FROM 'LSTAD' TO THE BOTTOM OF THE RT-11/XXDP MONITOR
                                        ;    IS FIRST WRITTEN WITH ITS ADDRESS AS DATA THEN READ BACK & VERIFIED.
                                        ;;*****************************************************************

004204  004737  012136          MEMTST: JSR     PC,EXAMKB

004210  012737  013176  000004          MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
004216  012737  000200  000006          MOV     #200,ERRVEC+2

004224  013703  012022                  MOV     MAXMEM,R3
004230  162703  000000                  SUB     #0,R3           ;TEST ONLY AS FAR AS XXDP MONITOR IF NOT RT-11
                                                                ;WHEN XXDP AVAIL. SUBSTRACT 1.5K (10)

                                        ;;*****************************************************************
                                        ;TEST FOR TIMEOUT FROM 0 TO 'LSTAD'
                                        ;;*****************************************************************

004234  005004                  1$:     CLR     R4              ;ADDR POINTER
004236  005037  013204                  CLR     INTFLG
004242  005714                          TST     (R4)            ;TEST LOCS 0 THRU END OF PGM.
004244  005737  013204                  TST     INTFLG          ;TIMEOUT?
004250  001403                          BEQ     2$              ;BR IF NO
004252  010437  004546                  MOV     R4,ADDR         ;FOR ERR TYP
004256  104001                          ERROR   1               ;TIMEOUT ON READ
004260  005724                  2$:     TST     (R4)+           ;BUMP ADDR
004262  020427  027744                  CMP     R4,#LSTAD       ;AT END?
004266  001363                          BNE     1$              ;BR IF NO
```

```
1747                                    ;;*****************************************************************
1748                                    ; TEST READ/WRITE PATTERNS FROM 'LSTAD' TO BOTTOM OF MONITOR
1749                                    ;;*****************************************************************
1750
1751  004270  005005                            CLR     R5              ;PASS CTR
1752  004272  012737  125252  004544            MOV     #125252,PAT     ;DATA PATT FOR 1'ST PASS
1753  004300  012704  027744           8$:     MOV     #LSTAD,R4       ;ADDR POINTER. R/W LOCS LSTAD THRU
1754                                                                    ;BOT OF RT11 OR BOT OF ABS LOADER
1755  004304  005037  013204           3$:     CLR     INTFLG
1756  004310  013714  004544                    MOV     PAT,(R4)        ;WRITE
1757  004314  005737  013204                    TST     INTFLG          ;TIMEOUT?
1758  004320  001403                            BEQ     4$              ;BR IF NO
1759  004322  010437  004546                    MOV     R4,ADDR         ;FOR ERR TYP
1760  004326  104002                            ERROR   2               ;TIMEOUT ON WRITE
1761  004330  011437  004542           4$:     MOV     (R4),MEMHLD     ;READ
1762  004334  023737  004542  004544            CMP     MEMHLD,PAT      ;COMPARE
1763  004342  001403                            BEQ     6$
1764  004344  010437  004546                    MOV     R4,ADDR         ;FOR ERR TYP
1765  004350  104003                            ERROR   3               ;COMPARE ERROR
1766
1767  004352  005724                    6$:     TST     (R4)+           ;BUMP POINTER
1768  004354  004737  004510                    JSR     PC,BOTMON       ;SEE IF AT BOTTOM OF MONITOR
1769  004360  000751                            BR      3$              ;RET HERE IF NO
1770
1771  004362  005705                            TST     R5              ;ELSE HERE
1772  004364  001005                            BNE     9$
1773  004366  005205                            INC     R5
1774  004370  012737  052525  004544            MOV     #052525,PAT     ;DATA PATT FOR 2'ND PASS
1775  004376  000740                            BR      8$              ;REPEAT
1776
1777                                    ;;*****************************************************************
1778                                    ;ADDRESS TEST. WRITE EACH LOC. WITH ITS ADDR. AS DATA. THEN VERIFY.
1779                                    ;;*****************************************************************
1780
1781  004400  012704  027744           9$:     MOV     #LSTAD,R4       ;ADDR REG
1782
1783  004404  010414                    10$:    MOV     R4,(R4)         ;LOAD ADDR AS DATA
1784  004406  005724                            TST     (R4)+           ;BUMP ADDR
1785  004410  004737  004510                    JSR     PC,BOTMON       ;SEE IF AT BOT OF MON
1786  004414  000773                            BR      10$             ;RET HERE IF NO
1787
1788                                                                    ;VERIFY PASS
1789  004416  012704  027744                    MOV     #LSTAD,R4
1790  004422  020414                    11$:    CMP     R4,(R4)         ;READ & CHECK
1791  004424  001407                            BEQ     12$             ;BR IF OK
1792  004426  010437  004546                    MOV     R4,ADDR
1793  004432  010437  004544                    MOV     R4,PAT
1794  004436  011437  004542                    MOV     (R4),MEMHLD
1795  004442  104005                            ERROR   5               ;MISCOMPARE
1796
1797  004444  005724                    12$:    TST     (R4)+           ;BUMP ADDR
1798  004446  004737  004510                    JSR     PC,BOTMON       ;SEE IF AT BOT OF MON
1799  004452  000763                            BR      11$             ;RET HERE IF NO
1800
1801  004454  012737  000006  000004            MOV     #ERRVEC+2,ERRVEC        ;RESTORE TIMEOUT VECTOR
1802  004462  005037  000006                    CLR     ERRVEC+2
```

```
 1803
 1804  004466  032777  010000  174444          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
 1805  004474  001426                          BEQ     EISFIS
 1806  004476  104401  020776                  TYPE    ,MEMORY
 1807  004502  104401  021011                  TYPE    ,DUN
 1808  004506  000421                          BR      EISFIS
 1809
```

```
1810                                    ;;*****************************************************************
1811                                    ;ROUTINE TO DETERMINE IF AT BOTTOM OF RT-11 OR XXDP MONITOR
1812                                    ;;*****************************************************************
1813
1814  004510  023727  004550  001000  BOTMON: CMP     HLD42,#1000     ;RT-11 IF LOC 42 = 1000
1815  004516  001403                          BEQ     1$              ;BR IF RT-11
1816  004520  020403                          CMP     R4,R3           ;ELSE AT BOT OF XXDP MON?
1817  004522  001006                          BNE     3$              ;BR IF NO
1818  004524  000403                          BR      2$
1819
1820  004526  020437  000054          1$:     CMP     R4,54           ;LOC 54 CONTAINS LOW ADDR OF RT-11 MON.
1821  004532  001002                          BNE     3$              ;BR IF NOT THERE
1822  004534  062716  000002          2$:     ADD     #2,(SP)         ;RET+2 IF AT END
1823  004540  000207                  3$:     RTS     PC
1824
1825  004542  000000                  MEMHLD: 0               ;PUT MEMORY READ HERE
1826  004544  000000                  PAT:    0               ;PATT TO BE WRITTEN & READ
1827  004546  000000                  ADDR:   0               ;ADDR UNDER TEST
1828  004550  000000                  HLD42:  0               ;SAVE LOC 42
```

```
1829                                    ;;*****************************************************************
1830                                    ;               EIS/FIS TEST SECTION
1831                                    ;;*****************************************************************
1832  004552  012737  001750  005424  EISFIS: MOV     #1000.,EISCNT   ;LOAD EXECUTION LOOP COUNTER
1833  004560  005737  012012                 TST     EIPRES          ;TEST IF SIZER DETERMINES EIS-FIS PRESENT
1834  004564  001404                         BEQ     1$              ;BR IF NOT PRESENT
1835  004566  032737  000100  001246         BIT     #BIT6,$DFVM     ;TEST IF EIS/FIS IS ENABLED
1836  004574  001402                         BEQ     ASH1            ;BR IF YES
1837  004576  000137  005446           1$:   JMP     NOEIS           ;BYPASS THIS SECTION - NOT SELECTED
1838                                    ;;*****************************************************************
1839                                    ;TEST ASH LEFT INSTRUCTION
1840                                    ;;*****************************************************************
1841  004602  012702  125252           ASH1:   MOV     #125252,R2      ;LOAD VALUE TO BE SHIFTED
1842  004606  000257                         CCC                     ;CLEAR CONDITION CODES
1843  004610  000264                         SEZ                     ;SET Z BIT
1844  004612  000270                         SEN                     ;SET N BIT
1845  004614  072227  000001                 ASH     #1,R2           ;SHIFT LEFT 1, RESULT = 52524
1846                                                                  ;N=0, Z=0, V=1, C=1
1847  004620  100403                         BMI     1$              ;IF N SET, REPORT ERROR
1848  004622  001402                         BEQ     1$              ;IF Z SET, REPORT ERROR
1849  004624  102001                         BVC     1$              ;IF V = 0, REPORT ERROR
1850  004626  103401                         BCS     2$              ;IF C SET, GO ON WITH TEST
1851  004630  104126           1$:   ERROR   126             ;INCORRECT CONDITION CODES FOR "ASH LEFT"
1852  004632  022702  052524           2$:   CMP     #52524,R2       ;RESULT CORRECT ?
1853  004636  001401                         BEQ     ASH2            ;BR IF OK
1854  004640  104127                         ERROR   127             ;INCORRECT DATA FOR "ASH LEFT"
1855                                    ;;*****************************************************************
1856                                    ;TEST ASH RIGHT INSTRUCTION
1857                                    ;;*****************************************************************
1858  004642  012701  052525           ASH2:   MOV     #52525,R1       ;LOAD VALUE TO BE SHIFTED
1859  004646  000257                         CCC                     ;CLEAR CONDITION CODES
1860  004650  072127  177777                 ASH     #-1,R1          ;SHIFT RIGHT 1, RESULT = 25252
1861  004654  022701  025252                 CMP     #25252,R1       ;RESULT CORRECT ?
1862  004660  001401                         BEQ     ASH3            ;BR IF OK
1863  004662  104127                         ERROR   127             ;INCORRECT DATA FOR "ASH RIGHT"
1864                                    ;;*****************************************************************
1865                                    ;TEST ASH LEFT SHIFT 000001 BY 16 SETS C.
1866                                    ;;*****************************************************************
1867  004664  012700  000001           ASH3:   MOV     #1,R0           ;LOAD VALUE TO BE SHIFTED
1868  004670  000241                         CLC                     ;CLEAR C BIT
1869  004672  072027  000020                 ASH     #16.,R0         ;SHIFT LEFT 16 TIMES
1870  004676  103001                         BCC     1$              ;BR IF ERROR
1871  004700  001401                         BEQ     ASH4            ;BR IF DATA CORRECT
1872  004702  104126           1$:   ERROR   126             ;INCORRECT CONDITION CODES FOR "ASH LEFT"
1873                                    ;;*****************************************************************
1874                                    ;TEST ASH RIGHT SHIFT 100000 BY 16 SETS C.
1875                                    ;;*****************************************************************
1876  004704  012700  100000           ASH4:   MOV     #100000,R0      ;LOAD VALUE TO BE SHIFTED
1877  004710  000241                         CLC                     ;CLEAR C BIT
1878  004712  072027  177760                 ASH     #-16.,R0        ;SHIFT RIGHT 16 TIMES
1879  004716  103401                         BCS     1$              ;BR IF NO ERROR
1880  004720  104126                         ERROR   126             ;INCORRECT CONDITION CODE FOR "ASH RIGHT"
1881  004722  020027  177777           1$:   CMP     R0,#177777      ;CHECK DATA
1882  004726  001401                         BEQ     ASHC1           ;BR IF OK
1883  004730  104127                         ERROR   127             ;INCORRECT DATA FOR "ASH RIGHT"
```

```
1884
1885                                    ;;*****************************************************************
1886                                    ;TEST ASHC LEFT INSTRUCTION
1887                                    ;;*****************************************************************
1888  004732  005002                    ASHC1:  CLR     R2              ;CLEAR MSW REGISTER
1889  004734  012703  125252                    MOV     #125252,R3      ;LOAD VALUE TO BE SHIFTED
1890  004740  000257                            CCC                     ;CLEAR CONDITION CODES
1891  004742  000264                            SEZ                     ;SET Z BIT
1892  004744  000261                            SEC                     ;SET C BIT
1893  004746  073227  000020                    ASHC    #16.,R2         ;SHIFT LEFT R3 INTO R2
1894                                                                    ;RESULT: N=1,Z=0,V=1,C=0
1895  004752  100003                            BPL     1$              ;IF N = 0, REPORT ERROR
1896  004754  001402                            BEQ     1$              ;IF Z SET, REPORT ERROR
1897  004756  102001                            BVC     1$              ;IF V - 0, REPORT ERROR
1898  004760  103001                            BCC     2$              ;IF C = 0, OK
1899  004762  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR ASHC LEFT
1900  004764  022702  125252            2$:     CMP     #125252,R2      ;R2 OK
1901  004770  001401                            BEQ     3$              ;BR IF CORRECT
1902  004772  104127                            ERROR   127             ;INCORRECT DATA FOR "ASHC LEFT"
1903  004774  005703                    3$:     TST     R3              ;R3 OK
1904  004776  001401                            BEQ     ASHC2           ;BR IF CORRECT
1905  005000  104127                            ERROR   127             ;INCORRECT DATA FOR "ASHC LEFT"
1906
1907                                    ;;*****************************************************************
1908                                    ;TEST ASHC RIGHT INSTRUCTION
1909                                    ;;*****************************************************************
1910  005002  005003                    ASHC2:  CLR     R3              ;CLEAR MSW REGISTER
1911  005004  012702  125252                    MOV     #125252,R2      ;LOAD VALUE TO BE SHIFTED
1912  005010  000257                            CCC                     ;CLEAR CONDITION CODES
1913  005012  000264                            SEZ                     ;SET Z BIT
1914  005014  000262                            SEV                     ;SET V BIT
1915  005016  000261                            SEC                     ;SET C BIT
1916  005020  073227  177760                    ASHC    #-16.,R2        ;SHIFT RIGHT 16 PLACES
1917                                                                    ;RESULT: N=1,Z=V=C=0
1918  005024  100002                            BPL     1$              ;IF N =0,REPORT ERROR
1919  005026  101401                            BLOS    1$              ;IF C OR Z SET, REPORT ERROR
1920  005030  102001                            BVC     2$              ;IF V = 0, GO ON WITH TEST
1921  005032  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODES FOR "ASHC RIGHT"
1922  005034  022702  177777            2$:     CMP     #-1,R2          ;TEST R2
1923  005040  001401                            BEQ     3$              ;BR IF CORRECT
1924  005042  104127                            ERROR   127             ;INCORRECT DATA FOR "ASHC RIGHT"
1925  005044  022703  125252            3$:     CMP     #125252,R3      ;TEST R2 TO R3 SHIFT
1926  005050  001401                            BEQ     ASHC3           ;BR IF CORRECT
1927  005052  104127                            ERROR   127             ;INCORRECT DATA FOR "ASHC RIGHT"
```

```
1928                                    ;;*****************************************************************
1929                                    ;TEST ASHC LEFT SHIFT 00000,0002 BY 31 SETS C.
1930                                    ;;*****************************************************************
1931  005054  005000                    ASHC3:  CLR     R0              ;CLEAR MSW REGISTER
1932  005056  012701  000002                    MOV     #2,R1           ;LOAD VALUE TO BE SHIFTED
1933  005062  073027  000037                    ASHC    #31.,R0         ;LEFT SHIFT 0002 BY 31.
1934  005066  103001                            BCC     1$              ;BR IF C BIT CLEARED
1935  005070  001401                            BEQ     ASHC4           ;BR IF RESULT =0
1936  005072  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR "ASHC LEFT"
1937
1938
1939                                    ;;*****************************************************************
1940                                    ;TEST ASHC RIGHT SHIFT 100000,00000 BY 32 SETS C.
1941                                    ;;*****************************************************************
1942  005074  012700  100000            ASHC4:  MOV     #100000,R0      ;LOAD VALUE TO BE SHIFTED
1943  005100  005001                            CLR     R1              ;CLEAR LSW REGISTER
1944  005102  073027  000040                    ASHC    #32.,R0         ;SHIFT RIGHT BY 32.
1945  005106  103001                            BCC     1$              ;BR IF C BIT CLEARED
1946  005110  100401                            BMI     2$              ;BR IF N BIT SET
1947  005112  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR "ASHC RIGHT"
1948  005114  022701  177777            2$:     CMP     #-1,R1          ;CHECK LOW ORDER RESULT
1949  005120  001003                            BNE     3$              ;BR IF IN ERROR
1950  005122  022700  177777                    CMP     #-1,R0          ;CHECK HIGH ORDER RESULT
1951  005126  001401                            BEQ     MUL0            ;BR IF CORRECT
1952  005130  104127                    3$:     ERROR   127             ;INCORRECT DATA FOR "ASHC RIGHT"
```

```
1953                                    ;;*****************************************************************
1954                                    ;TEST MUL INSTRUCTION
1955                                    ;;*****************************************************************
1956                                    ;       125252*40000=165252,100000
1957  005132  012700  125252   MUL0:    MOV     #125252,R0      ;LOAD MULTIPLICAND
1958  005136  070027  040000            MUL     #40000,R0       ;R1:R0=40000*125252
1959  005142  100003                    BPL     1$              ;BR IF N BIT ERROR
1960  005144  102402                    BVS     1$              ;BR IF V BIT ERROR
1961  005146  001401                    BEQ     1$              ;BR IF Z BIT ERROR
1962  005150  103401                    BCS     2$              ;BR IF C BIT CORRECT
1963  005152  104126            1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR "MUL"
1964  005154  022700  165252    2$:     CMP     #165252,R0      ;HIGH ORDER RESULT OK ?
1965  005160  001003                    BNE     3$              ;BR IF NO
1966  005162  022701  100000            CMP     #100000,R1      ;LOW ORDER RESULT OK?
1967  005166  001401                    BEQ     DIV0            ;BR IF YES
1968  005170  104127            3$:     ERROR   127             ;INCORRECT DATA FOR "MUL"
1969
1970
1971                                    ;;*****************************************************************
1972                                    ;TEST DIV INSTRUCTION
1973                                    ;;*****************************************************************
1974  005172  012701  125252   DIV0:    MOV     #125252,R1      ;LOAD LOW ORDER DIVIDEND
1975  005176  012700  177777            MOV     #-1,R0          ;HIGH ORDER DIVIDEND
1976  005202  000261                    SEC                     ;SET C BIT TO 1
1977  005204  071027  000002            DIV     #2,R0           ;R1:R0=177777,125252/2
1978  005210  103403                    BCS     1$              ;BR IF C BIT ERROR
1979  005212  102402                    BVS     1$              ;BR IF V BIT ERROR
1980  005214  001401                    BEQ     1$              ;BR IF Z BIT ERROR
1981  005216  100401                    BMI     2$              ;BR IF N BIT OK
1982  005220  104126            1$:     ERROR   126             ;INCORRECT DONDITION CODE FOR "DIV"
1983  005222  005701            2$:     TST     R1              ;CHECK LOW ORDER RESULT
1984  005224  001003                    BNE     3$              ;BR IF ERROR
1985  005226  020027  152525            CMP     R0,#152525      ;CHECK HIGH ORDER RESULT
1986  005232  001401                    BEQ     FIS             ;BR IF OK
1987  005234  104127            3$:     ERROR   127             ;INCORRECT DATA FOR "DIV"
```

```
1988
1989                            ;;*****************************************************************
1990                            ;TEST ALL FLOATING INSTRUCTIONS TOGETHER
1991                            ;                                              045670,123456
1992                            ;       134343,107070 * 032107,065432 * ----------------------------
1993                            ;                                       (135252,125252 - 040616,016161)
1994                            ;
1995                            ;       THE RESULT =  137201,115230
1996                            ;;*****************************************************************
1997 005236 012704 027744       FIS:    MOV     #LSTAD,R4               ;LOAD R4 STACK POINTER
1998 005242 012744 107070               MOV     #107070,-(R4)           ;LOAD DATA ONTO STACK
1999 005246 012744 134343               MOV     #134343,-(R4)
2000 005252 012744 065432               MOV     #065432,-(R4)
2001 005256 012744 032107               MOV     #032107,-(R4)
2002 005262 012744 123456               MOV     #123456,-(R4)
2003 005266 012744 045670               MOV     #045670,-(R4)
2004 005272 012744 125252               MOV     #125252,-(R4)
2005 005276 012744 135252               MOV     #135252,-(R4)
2006 005302 012744 016161               MOV     #016161,-(R4)
2007 005306 012744 040616               MOV     #040616,-(R4)
2008 005312 000257                      CCC
2009 005314 000264                      SEZ                             ;SET Z BIT
2010 005316 000240                      NOP
2011 005320 075014                      FSUB    R4      ;135252,125252-040616,016161=140616,017434
2012 005322 075034                      FDIV    R4      ;045670,123456/140616,017434=145246,047065
2013 005324 075024                      FMUL    R4      ;032107,065432*145246,047065=137201,106137
2014 005326 075004                      FADD    R4      ;134343,107070+137201,106137=137201,115230
2015 005330 103403                      BCS     1$              ;BR IF C SET, REPORT ERROR
2016 005332 102402                      BVS     1$              ;BR IF V SET, REPORT ERROR
2017 005334 001401                      BEQ     1$              ;BR IF Z SET, REPORT ERROR
2018 005336 100401                      BMI     2$              ;BR IF N SET, DO NEXT TEST
2019 005340 104126              1$:     ERROR   126             ;INCORRECT CONDITION CODE AFTER "FADD,FSUB,FMUL,FDIV"
2020 005342 012437 005420       2$:     MOV     (R4)+,ANS1      ;SAVE FIRST HALF OF THE ANSWER
2021 005346 012437 005422               MOV     (R4)+,ANS2      ;SAVE 2ND HALF
2022 005352 020427 027744               CMP     R4,#LSTAD       ;CHECK R4 STACK POINTER
2023 005356 001401                      BEQ     3$              ;BR IF CORRECT
2024 005360 104127                      ERROR   127             ;INCORRECT STACK POINTER VALUE
2025 005362 022737 137201 005420 3$:    CMP     #137201,ANS1    ;TEST 1ST HALF OF RESULT
2026 005370 001401                      BEQ     4$              ;BR IF CORRECT
2027 005372 104127                      ERROR   127             ;INCORRECT DATA FROM A "FIS INSTRUCTION"
2028 005374 022737 115230 005422 4$:    CMP     #115230,ANS2    ;TEST 2ND HALF OF RESULT
2029 005402 001401                      BEQ     5$              ;BR IF CORRECT
2030 005404 104127                      ERROR   127             ;INCORRECT DATA FORM A "FIS INSTRUCTION"
2031 005406 005337 005424       5$:     DEC     EISCNT          ;FINISHED THE # OF EXECUTION LOOPS ?
2032 005412 001405                      BEQ     EISDON          ;BR IF FINISHED
2033 005414 000137 004602               JMP     ASH1            ;LOOP UNTIL DONE
2034 005420 000000              ANS1:   0
2035 005422 000000              ANS2:   0
2036 005424 000000              EISCNT: 0
2037
2038 005426 032777 010000 173504 EISDON: BIT    #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2039 005434 001404                      BEQ     NOEIS           ;BR IF CLEARED
2040 005436 104401 020757               TYPE    ,FISEIS         ;REPORT EIS/FIS TESTING COMPLETED
2041 005442 104401 021011               TYPE    ,DUN            ;REPORT "DONE"
2042 005446                     NOEIS:                  ;TAG TO BYPASS THE EIS/FIS TEST SECTION
```

```
2042
2043                                    ;;*****************************************************************
2044                                    ;       START LINE CLOCK
2045                                    ;;*****************************************************************
2046
2047  005446  032737  000200  001246  CLKTST: BIT     #BIT7,$DEVM     ;DROP TEST?
2048  005454  001031                          BNE     PRTST           ;BR IF YES
2049  005456  012737  013176  000100          MOV     #INTSRV,CLKVEC  ;SETUP VECTOR AREA
2050  005464  012737  000200  000102          MOV     #200,CLKVEC+2   ;LOCKOUT OTHER INTER.
2051  005472  005037  013204                  CLR     INTFLG
2052  005476  052737  000100  177546          BIS     #IE,CLK         ;SET CLOCK LOOSE!
2053
2054  005504  004537  012352                  JSR     R5,TIMER        ;SEE IF INTFLG GOES TO 100
2055  005510  013204                            INTFLG
2056  005512  000100                            100
2057  005514  104032                          ERROR   32              ;CLK NOT RESPONDING
2058  005516  000410                          BR      PRTST
2059
2060  005520  032777  010000  173412          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2061  005526  001404                          BEQ     PRTST
2062  005530  104401  020300                  TYPE    ,CLOCK
2063  005534  104401  020704                  TYPE    ,RUN
```

```
2064
2065                                    ;;*****************************************************************
2066                                    ;       START PRINTER
2067                                    ;;*****************************************************************
2068
2069  005540  032737  100000  001246  PRTST:  BIT     #BIT15,$DEVM    ;DROP TEST?
2070  005546  001056                          BNE     SMTST           ;BR IF YES
2071  005550  005737  012016                  TST     PRPRES          ;PRINTER THERE?
2072  005554  001453                          BEQ     SMTST           ;BR IF NO
2073
2074  005556  012737  027014  177420  2$:     MOV     #<PRT!B9600!CHAR8>,PARAM        ;SETUP BAUD RATE & CHAR LENGTH
2075  005564  012737  013206  000200          MOV     #PRSRV,PRVEC    ;ELSE SETUP PRINTER VECTOR
2076  005572  012737  000200  000202          MOV     #200,PRVEC+2    ;LOCKOUT INTERR
2077
2078  005600  005037  013334                  CLR     LINCT           ;CLEAR LINE CTR
2079  005604  012737  000015  013332          MOV     #CR,PRCHR       ;CHAR TO BE PRINTED
2080                                                                  ;'DEL' (177) CAN BE USED TO CLR PRINTER BUFF
2081  005612  052737  00010C  177514          BIS     #IE,PRS         ;SET PRINTER LOOSE!
2082                                                                  ;WILL PRINT CONTINUOUS 132 COLUMN LINES
2083                                                                  ;EA BEGINNING WITH ASCII 40 THRU 176
2084  005620  004537  012352                  JSR     R5,TIMER        ;SEE IF LINCT GOES TO 5
2085  005624  013334                            LINCT
2086  005626  000005                            5
2087  005630  104033                          ERROR   33              ;PRINTER NOT RESPONDING
2088  005632  000424                          BR      SMTST
2089
2090  005634  005737  012020                  TST     PRPORT          ;EXT LOOPBACK MODE?
2091  005640  001403                          BEQ     3$              ;BR IF NO
2092  005642  042737  000100  177514          BIC     #IE,PRS         ;ELSE SHUT DOWN PRINTER
2093
2094  005650  032777  010000  173262  3$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2095  005656  001412                          BEQ     SMTST
2096  005660  104401  020240                  TYPE    ,PRINT
2097  005664  005737  012020                  TST     PRPORT          ;EXT LOOPBACK MODE?
2098  005670  001403                          BEQ     4$              ;BR IF NO
2099  005672  104401  020251                  TYPE    ,PORT
2100  005676  000402                          BR      SMTST
2101  005700  104401  020704          4$:     TYPE    ,RUN
```

```
2102
2103                                    ;;*****************************************************************
2104                                    ;       START SYNC COMM PORT
2105                                    ;;*****************************************************************
2106
2107  005704  032737  002000  001246  SMTST:  BIT     #BIT10,$DEVM    ;DROP TEST?
2108  005712  001402                          BEQ     8$              ;BR IF NO
2109  005714  000137  006160                  JMP     AMTST           ;ELSE JUMP TEST
2110
2111  005720  123727  001210  000001  8$:     CMPB    $ENV,#1         ;UNDER APT?
2112  005726  001003                          BNE     9$              ;BR IF NO
2113  005730  005737  001176                  TST     $PASS           ;ELSE 1'ST PASS?
2114  005734  001111                          BNE     AMTST           ;EXIT TEST IF NO
2115
2116  005736  005037  014026          9$:     CLR     LSTCHR          ;LAST CHAR FLAG FOR XMIT
2117  005742  052737  000400  176624          BIS     #MR,SMXC        ;MASTER RESET
2118  005750  052737  004000  176624          BIS     #MCLK,SMXC      ;MAINT CLOCK
2119
2120  005756  032737  000010  001246          BIT     #BIT3,$DEVM     ;EXT LOOPBACK?
2121  005764  001403                          BEQ     1$              ;BR IF YES
2122  005766  052737  014000  176624          BIS     #<MM!MCLK>,SMXC ;ELSE SET MAINT MODE FOR INT. LOOPBACK
2123  005774  012737  007026  176622  1$:     MOV     #<CHR8!ODDPAR!026>,SMPAR        ;SETUP SYNC PARAMETER REGISTER
2124  006002  052737  000026  176620          BIS     #<RTS!DTR!SRSYN>,SMRC   ;SET REQ TO SEND, DATA TERM RDY
2125                                                                          ;& SEARCH SYNC
2126  006010  052737  000020  176624          BIS     #SEND,SMXC      ;ENABLE XMIT SEND
2127  006016  012737  000026  176626          MOV     #026,SMXB       ;XMIT SYNC CHAR
2128
2129  006024  004537  012634                  JSR     R5,TIMER2       ;WAIT FOR DONE
2130  006030  176620                              SMRC
2131  006032  000200                              DONE
2132  006034  104045                          ERROR   45              ;NO DONE
2133  006036  000435                          BR      5$              ;EXIT
2134
2135  006040  013737  176622  013736          MOV     SMRB,COMHLD     ;READ WORD
2136  006046  023727  013736  000026          CMP     COMHLD,#026     ;IS IT SYNC CHAR?
2137  006054  001402                          BEQ     4$              ;BR IF YES
2138  006056  104006                          ERROR   6               ;DID NOT REC SYNC CHAR
2139  006060  000424                          BR      5$              ;EXIT
2140
2141  006062  004537  012326          4$:     JSR     R5,SETVEC       ;SETUP VECTOR AREA
2142  006066  000340                              SMRVEC
2143  006070  014030                              SMRSRV
2144  006072  013740                              SMXSRV
```

```
 2145
 2146  006074  012737  000027  013734          MOV     #C27,COMCHR     ;1'ST CHAR TO BE XMITTED
 2147  006102  052737  000100  176620          BIS     #IE,SMRC        ;SET SYNC MODEM LOOSE!
 2148  006110  052737  000100  176624          BIS     #IE,SMXC
 2149
 2150  006116  004537  012352                  JSR     R5,TIMER        ;SEF IF COMCHR GOES TO ALL 1'S (DONE)
 2151  006122  013734                            COMCHR
 2152  006124  177777                            177777
 2153  006126  104037                          ERROR   37              ;SYNC COMM NOT RESPONDING
 2154  006130  000240                          NOP
 2155
 2156  006132  012737  000400  176624  5$:     MOV     #MR,SMXC        ;MASTER RESET
 2157  006140  032777  010000  172772          BIT     #BIT12,@SWR     ;DO TYPEOUT IF SET
 2158  006146  001404                          BEQ     AMTST           ;EXIT IF NOT
 2159  006150  104401  020211                  TYPE    ,SCOM
 2160  006154  104401  021011                  TYPE    ,DUN
```

```
2161
2162                                    ;;*****************************************************************
2163                                    ;       START ASYNC COMM PORT
2164                                    ;;*****************************************************************
2165
2166  006160  032737  004000  001246  AMTST:  BIT     #BIT11,$DEVM    ;DROP TEST?
2167  006166  001054                          BNE     CL1TST          ;BR IF YES
2168
2169  006170  032737  000010  001246          BIT     #BIT3,$DEVM     ;EXT LOOPBACK?
2170  006176  001404                          BEQ     1$              ;BR IF YES
2171  006200  012737  037036  177420          MOV     #<AMOD!B9600!OPAR!CHAR8!MAINT>,PARAM    ;ELSE SET FOR INT LOOPBACK
2172  006206  000403                          BR      3$
2173  006210  012737  037034  177420  1$:     MOV     #<AMOD!B9600!OPAR!CHAR8>,PARAM  ;WAKE UP AMOD IF JUST DID SMOD
2174  006216  052737  000006  176610  3$:     BIS     #<RTS!DTR>,AMRC ;SET REQ TO SEND & DATA TERM RDY
2175
2176  006224  004537  012326                  JSR     R5,SETVEC       ;SETUP VECTOR AREA
2177  006230  000330                              AMRVEC
2178  006232  013656                              AMRSRV
2179  006234  013640                              AMXSRV
2180
2181  006236  005037  013734                  CLR     COMCHR          ;CHAR TO BE XMITTED
2182  006242  013737  176612  013736          MOV     AMRB,COMHLD     ;CLR OUT ANY PREVIOUS STORED WORD
2183  006250  052737  000100  176610          BIS     #IE,AMRC        ;SET ASYNC MODEM LOOSE!
2184  006256  052737  000100  176614          BIS     #IE,AMXC
2185
2186  006264  004537  012352                  JSR     R5,TIMER        ;SEE IF COMCHR GOES TO 377
2187  006270  013734                            COMCHR
2188  006272  000377                            377
2189  006274  104040                          ERROR   40              ;ASYNC COMM NOT RESPONDING
2190  006276  000410                          BR      CL1TST
2191
2192  006300  032777  010000  172632          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2193  006306  001404                          BEQ     CL1TST
2194  006310  104401  020224                  TYPE    ,ACOM
2195  006314  104401  020704                  TYPE    ,RUN
```

```
2196
2197                                    ;;*****************************************************************
2198                                    ;       START CLUSTER TERMINAL #1
2199                                    ;;*****************************************************************
2200
2201  006320  005737  012014           CL1TST: TST     CLPRES          ;OPTION PRESENT?
2202  006324  001002                           BNE     1$              ;BR IF YES
2203  006326  000137  006774                   JMP     DXTST0
2204
2205  006332  032737  004000  001246  1$:     BIT     #4000,$DEVM     ;DROP ASYNC?
2206  006340  001406                           BEQ     4$              ;BR IF NO
2207  006342  042737  000006  176610           BIC     #<RTS!DTR>,AMRC ;MUST BE TOGGLED
2208  006350  052737  000006  176610           BIS     #<RTS!DTR>,AMRC ;PRIME IT
2209
2210  006356  032737  010000  001246  4$:     BIT     #BIT12,$DEVM    ;DROP TEST?
2211  006364  001051                           BNE     CL2TST          ;BR IF YES
2212
2213  006366  032737  000001  001246           BIT     #BIT0,$DEVM     ;TERM #1 SET FOR EXT LOOPBACK?
2214  006374  001404                           BEQ     2$              ;BR IF YES
2215  006376  012737  005016  177420           MOV     #<CT1!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
2216  006404  000403                           BR      3$
2217  006406  012737  005014  177420  2$:     MOV     #<CT1!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK
2218
2219  006414  004537  012326           3$:     JSR     R5,SETVEC       ;SETUP INTERR VECTORS
2220  006420  000300                               TK1VEC
2221  006422  013356                               TK1SRV
2222  006424  013340                               TP1SRV
2223
2224  006426  005037  013434                   CLR     T1CHR           ;CHAR TO BE XMITTED
2225  006432  013737  176502  013436           MOV     TK1B,T1HLD      ;READ CHAR (CLEAR STALE DATA)
2226  006440  052737  000100  176500           BIS     #IE,TK1S        ;SET CLUSTER TERM #1 LOOSE!
2227  006446  052737  000100  176504           BIS     #IE,TP1S
2228
2229  006454  004537  012352                   JSR     R5,TIMER        ;SEE IF T1CHR GOES TO 377
2230  006460  013434                             T1CHR
2231  006462  000377                             377
2232  006464  104034                           ERROR   34              ;CLUSTER TERM 1 NOT RESPONDING
2233  006466  000410                           BR      CL2TST
2234
2235  006470  032777  010000  172442           BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2236  006476  001404                           BEQ     CL2TST
2237  006500  104401  020156                   TYPE    ,CLT1
2238  006504  104401  020704                   TYPE    ,RUN
```

```
 2239
 2240                                   ;;*****************************************************************
 2241                                   ;       START CLUSTER TERMINAL #2
 2242                                   ;;*****************************************************************
 2243
 2244  006510  032737  020000  001246  CL2TST: BIT     #BIT13,$DEVM    ;DROP TEST?
 2245  006516  001051                         BNE     CL3TST          ;BR IF YES
 2246
 2247  006520  032737  000002  001246         BIT     #BIT1,$DEVM     ;TERM #2 SET FOR EXT LOOPBACK?
 2248  006526  001404                         BEQ     1$              ;BR IF YES
 2249  006530  012737  055016  177420         MOV     #<CT2!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
 2250  006536  000403                         BR      2$
 2251  006540  012737  055014  177420  1$:    MOV     #<CT2!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK
 2252
 2253  006546  004537  012326          2$:    JSR     R5,SETVEC       ;SETUP INTERR VECTORS
 2254  006552  000310                             TK2VEC
 2255  006554  013456                             TK2SRV
 2256  006556  013440                             TP2SRV
 2257
 2258  006560  005037  013534                 CLR     T2CHR           ;CHAR TO BE XMITTED
 2259  006564  013737  176512  013536         MOV     TK2B,T2HLD      ;READ CHAR (CLEAR STALE DATA)
 2260  006572  052737  000100  176510         BIS     #IE,TK2S        ;SET CLUSTER TERM #2 LOOSE!
 2261  006600  052737  000100  176514         BIS     #IE,TP2S
 2262
 2263  006606  004537  012352                 JSR     R5,TIMER        ;SEE IF T2CHR GOES TO 377
 2264  006612  013534                           T2CHR
 2265  006614  000377                           377
 2266  006616  104035                 ERROR   35              ;CLUSTER TERM 2 NOT RESPONDING
 2267  006620  000410                 BR      CL3TST
 2268
 2269  006622  032777  010000  172310         BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
 2270  006630  001404                         BEQ     CL3TST
 2271  006632  104401  020167                 TYPE    ,CLT2
 2272  006636  104401  020704                 TYPE    ,RUN
```

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

006642  032737  040000  001246  CL3TST: BIT     #BIT14,$DEVM    ;DROP TEST?
006650  001051                          BNE     DXTST0          ;BR IF YES

006652  032737  000004  001246          BIT     #BIT2,$DEVM     ;TERM #3 SET FOR EXT LOOPBACK?
006660  001404                          BEQ     1$              ;BR IF YES
006662  012737  065016  177420          MOV     #<CT3!B2400!CHAR8!MAINT>,PARAM  ;ELSE SET FOR INT LOOPBACK
006670  000403                          BR      2$
006672  012737  065014  177420  1$:     MOV     #<CT3!B2400!CHAR8>,PARAM        ;SETUP FOR EXT LOOPBACK

006700  004537  012326          2$:     JSR     R5,SETVEC       ;SETUP INTERR VECTORS
006704  000320                              TK3VEC
006706  013556                              TK3SRV
006710  013540                              TP3SRV

006712  005037  013634                  CLR     T3CHR           ;CHAR TO BE XMITTED
006716  013737  176522  013636          MOV     TK3B,T3HLD      ;READ CHAR (CLEAR STALE DATA)
006724  052737  000100  176520          BIS     #IE,TK3S        ;SET CLUSTER TERM #3 LOOSE!
006732  052737  000100  176524          BIS     #IE,TP3S

006740  004537  012352                  JSR     R5,TIMER        ;SEE IF T3CHR GOES TO 377
006744  013634                            T3CHR
006746  000377                            377
006750  104036                  ERROR   36              ;CLUSTER TERM 3 NOT RESPONDING
006752  000410                  BR      DXTST0

006754  032777  010000  172156          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
006762  001404                          BEQ     DXTST0
006764  104401  020200                  TYPE    ,CLT3
006770  104401  020704                  TYPE    ,RUN
```

```
                                        ;;*****************************************************************
                                        ;       FILL & EMPTY BUFFER TEST:  DRIVE 0
                                        ;
                                        ;A FILL BUFFER COMMAND IS ISSUED WITH ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
                                        ;AN EMPTY BUFFER COMMAND IS ISSUED TO FILL 'DOBUF'.
                                        ;UPON COMPLETION, DOBUF IS CHECKED FOR ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
                                        ;ERRORS IN THIS TEST MAY INDICATE CABLE CROSS-TALK INTERFERENCE.
                                        ;;*****************************************************************

006774  005037  003260          DXTST0: CLR     DXTST           ;0 = DXTST0

007000  032737  000400  001246          BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
007006  001404                          BEQ     7$              ;BR IF NO
007010  032737  001000  001246          BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
007016  001113                          BNE     DXTST1          ;BR IF YES

007020  004537  012634          7$:     JSR     R5,TIMER2       ;WAIT FOR DISK DONE
007024  177170                             RXCS
007026  000200                             DONE
007030  104075                          ERROR   75              ;NO DONE
007032  000500                          BR      8$              ;EXIT

007034  012737  000011  177170          MOV     #INITAL,RXCS    ;DO AN INITIALIZE COMMAND

007042  004537  012634                  JSR     R5,TIMER2       ;WAIT FOR DONE
007046  177170                             RXCS
007050  000200                             DONE
007052  104114                          ERROR   114             ;NO DONE
007054  000467                          BR      8$

007056  005737  177170                  TST     RXCS            ;ERROR?
007062  100001                          BPL     6$              ;BR IF NO
007064  104140                          ERROR   140             ;INIT CMD ERROR

                                                                ;FILL BUFFER
007066  012700  000100          6$:     MOV     #100,R0         ;WORD CT
007072  005001                          CLR     R1              ;WORD
007074  012737  000001  177170          MOV     #FBUF,RXCS      ;ISSUE FILL BUFF COMMAND
007102  010137  177172          1$:     MOV     R1,RXDB
007106  005101                          COM     R1              ;COMPLEMENT WORD
007110  005300                          DEC     R0              ;DONE ALL 100 WORDS?
007112  001373                          BNE     1$              ;BR IF NO

007114  004537  012634                  JSR     R5,TIMER2       ;WAIT FOR DISK DONE
007120  177170                             RXCS
007122  000200                             DONE
007124  104115                          ERROR   115             ;NO DONE
007126  000442                          BR      8$              ;EXIT
```

```
2356
2357  007130  012700  000100            MOV     #100,R0         ;WORD CT
2358  007134  012701  002564            MOV     #DOBUF,R1       ;BUFFER ADDR
2359  007140  012737  000003  177170    MOV     #EBUF,RXCS      ;ISSUE EMPTY BUFF COMMAND
2360  007146  013721  177172    2$:     MOV     RXDB,(R1)+      ;STORE WORD
2361  007152  005300                    DEC     R0              ;DONE?
2362  007154  001374                    BNE     2$              ;BR IF NO
2363
2364  007156  004537  012634            JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2365  007162  177170                       RXCS
2366  007164  000200                       DONE
2367  007166  104116                    ERROR   116             ;NO DONE
2368  007170  000421                    BR      8$              ;EXIT
2369
2370                                                            ;CHECK IT
2371  007172  012701  002564            MOV     #DOBUF,R1       ;BUFFER ADDR
2372  007176  012137  007240    3$:     MOV     (R1)+,EBHLD     ;STORE IT
2373  007202  001401                    BEQ     4$              ;BR IF ZERO
2374  007204  104076                    ERROR   76              ;WORD NOT ALL ZEROS
2375
2376  007206  012137  007240    4$:     MOV     (R1)+,EBHLD
2377  007212  023727  007240  177777    CMP     EBHLD,#-1       ;ALL 1'S?
2378  007220  001401                    BEQ     5$              ;BR IF YES
2379  007222  104077                    ERROR   77              ;WORD NOT ALL ONES
2380
2381  007224  020127  002764    5$:     CMP     R1,#DOBUF+200   ;END OF DOBUFF?
2382  007230  001362                    BNE     3$              ;BR IF NO
2383  007232  000405                    BR      DXTST1          ;GO TO NXT TST
2384
2385  007234  000137  011176    8$:     JMP     EOP             ;EXIT ALL DISK TESTS
2386
2387  007240  000000            EBHLD:  0                       ;STORAGE        FOR ERROR REPORT
2388  007242  000000            EXP0:   0                       ;EXPECT 0'S     FOR ERROR REPORT
2389  007244  177777            EXP1:   -1                      ;EXPECT 1'S     FOR ERROR REPORT
```

```
2390
2391                                    ;;*****************************************************************
2392                                    ;       START DX0 & DX1 DATA PATTERN
2393                                    ;
2394                                    ;EACH SECTOR IS FILLED WITH ITS TRACK & SECTOR ADDRESS AS DATA.
2395                                    ;DX1 WILL ALWAYS HAVE ITS DATA BIT 15 SET TO DISTINGUISH IT FROM DX0.
2396                                    ;EACH TRACK IS READ & CHECKED BEFORE GOING TO THE NEXT TRACK.
2397                                    ;DATA IS WRITTEN ON ALTERANTE SECTORS TO AVOID LATENCY TIMES.
2398                                    ;IE: SECTORS 1,3,5...31 FOLLOWED BY 2,4,6...32.
2399                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2400                                    ;1'ST PASS PERFORMED ON TRACKS 0-4, 40-44, 72-76(10)
2401                                    ;SUBSEQUENT PASSES ON ALL TRACKS
2402                                    ;;*****************************************************************
2403
2404  007246  012737  000001  003260  DXTST1: MOV     #1,DXTST        ;1 = DATA PATT TESTS
2405
2406  007254  005037  002546          CLR     D0TRK           ;STARTING TRACK
2407  007260  005037  003016          CLR     D1TRK
2408
2409  007264  005037  002540  12$:     CLR     D0PAT           ;INIT DX0 FLAGS
2410  007270  005037  002544          CLR     D0ERR
2411  007274  005037  002554          CLR     D0CERR
2412  007300  012737  000001  002550  MOV     #1,D0SEC
2413
2414  007306  005037  003010          CLR     D1PAT           ;INIT DX1 FLAGS
2415  007312  005037  003014          CLR     D1ERR
2416  007316  005037  003024          CLR     D1CERR
2417  007322  012737  000001  003020  MOV     #1,D1SEC
2418
2419  007330  032737  000400  001246  BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2420  007336  001031                  BNE     2$              ;BR IF YES
2421
2422  007340  012737  014130  000264  MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2423  007346  012737  000200  000266  MOV     #200,RXVEC+2
2424
2425  007354  005737  003274  1$:     TST     COMP2           ;DOING COMPAT PASS 2?
2426  007360  001404                  BEQ     10$             ;BR IF NO
2427  007362  012737  014622  014230  MOV     #D0RSEC,DODISP  ;ELSE DO RFADS ONLY
2428  007370  000403                  BR      7$
2429
2430  007372  012737  014246  014230  10$:    MOV     #D0FBUF,DODISP  ;DO NORMAL TESTING
2431  007400  005037  003262  7$:      CLR     FIN             ;DO DX1 WHEN SET
2432  007404  052737  000100  177170  BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE!
2433
2434  007412  004737  012436          JSR     PC,TIMDO
2435  007416  104041                  ERROR   41              ;DX0 NOT GETTING TRACK DONE FLAG
2436  007420  000240                  NOP
2437
2438  007422  032737  001000  001246  2$:      BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2439  007430  001031                  BNE     4$              ;BR IF YES
2440
2441  007432  012737  014130  000264  MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2442  007440  012737  000200  000266  MOV     #200,RXVEC+2
```

```
2443
2444  007446  005737  003274          3$:     TST     COMP2           ;DOING COMPAT PASS 2?
2445  007452  001404                          BEQ     11$             ;BR IF NO
2446  007454  012737  016406  014236          MOV     #D1RSEC,D1DISP  ;ELSE DO READ ONLY
2447  007462  000403                          BR      8$
2448
2449  007464  012737  016032  014236  11$:    MOV     #D1FBUF,D1DISP  ;DO NORMAL TESTING
2450  007472  005037  003262          8$:     CLR     FIN             ;GO BACK TO DX0 WHEN SET
2451  007476  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2452
2453  007504  004737  012532                  JSR     PC,TIMD1
2454  007510  104042                          ERROR   42              ;DX1 NOT GETTING TRACK DONE FLAG
2455  007512  000240                          NOP
2456
2457  007514  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2458  007522  001007                          BNE     5$              ;BR IF YES
2459  007524  005737  002540                  TST     DOPAT           ;ELSE IS DX0 ALL DONE?
2460  007530  001004                          BNE     5$              ;BR IF YES
2461  007532  005737  012530                  TST     DOTMO           ;TIMEOUT?
2462  007536  001706                          BEQ     1$              ;BR IF NO
2463  007540  000717                          BR      7$              ;ELSE CONT LAST COMMAND
2464
2465  007542  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2466  007550  001007                          BNE     6$              ;BR IF YES
2467  007552  005737  003010                  TST     D1PAT           ;ELSE IS DX1 ALL DONE?
2468  007556  001004                          BNE     6$              ;BR IF YES
2469  007560  005737  012620                  TST     D1TMO           ;TIMEOUT?
2470  007564  001730                          BEQ     3$              ;BR IF NO
2471  007566  000741                          BR      8$              ;ELSE CONT LAST COMMAND
2472
2473  007570  005737  003272          6$:     TST     COMP1           ;DOING COMPAT PASS 1?
2474  007574  001402                          BEQ     13$             ;BR IF NO
2475  007576  000137  011176                  JMP     EOP             ;ELSE ALL DONE
2476
2477  007602  005737  001176          13$:    TST     $PASS           ;1'ST PASS?
2478  007606  001035                          BNE     DXTST2          ;BR IF NO
2479
2480  007610  023727  003264  000004          CMP     MAXTRK,#4       ;JUST DID 5 OUTER TRACKS?
2481  007616  001012                          BNE     14$             ;BR IF NO
2482  007620  012737  000050  002546          MOV     #40.,DOTRK      ;ELSE  SETUP FOR 5 MIDDLE TRKS
2483  007626  012737  000050  003016          MOV     #40.,D1TRK
2484  007634  012737  000054  003264          MOV     #44.,MAXTRK
2485  007642  000610                          BP      12$             ;GO DO IT.
2486
2487  007644  023727  003264  000054  14$:    CMP     MAXTRK,#44.     ;JUST DID 5 MIDDLE TRACKS?
2488  007652  001013                          BNE     DXTST2          ;BR IF NO, MUST BF ALL DONE
2489  007654  012737  000110  002546          MOV     #72.,DOTRK      ;ELSE SETUP FOR 5 INNER TRKS
2490  007662  012737  000110  003016          MOV     #72.,D1TRK
2491  007670  012737  000114  003264          MOV     #76.,MAXTRK
2492  007676  000137  007264                  JMP     12$             ;DO IT'
2493
```

```
2494
2495                                    ;;*****************************************************************
2496                                    ;       START DX0 & DX1 RANDOM SEEKS
2497                                    ;
2498                                    ;RANDOM SEEKS ARE PERFORMED ON EACH DISK.
2499                                    ;DATA IS READ & VERIFIED TO BE CORRECT FROM THE PREVIOUS TEST.
2500                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2501                                    ;1'ST PASS PERFORMS 20 RANDOM SEEKS ON 1'ST 5 TRACKS.
2502                                    ;SUBSEQUENT PASSES PERFORMS 500 RANDOM SEEKS BETWEEN ALL TRACKS.
2503                                    ;;*****************************************************************
2504
2505  007702  012737  000002  003260  DXTST2: MOV     #2,DXTST        ;2 = RANDOM SEEK TESTS
2506  007710  005737  001176                  TST     $PASS           ;1'ST PASS?
2507  007714  001003                          BNE     7$              ;BR IF NO
2508  007716  012737  000004  003264          MOV     #4,MAXTRK       ;ELSE JUST DO TRKS 0-4
2509
2510  007724  005037  002544          7$:     CLR     D0ERR           ;INIT DX0 FLAGS
2511  007730  005037  002554                  CLR     D0CERR
2512  007734  005037  002542                  CLR     D0CNT
2513
2514  007740  005037  003014                  CLR     D1ERR           ;INIT DX1 FLAGS
2515  007744  005037  003024                  CLR     D1CERR
2516  007750  005037  003012                  CLR     D1CNT
2517
2518  007754  032737  000400  001246          BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2519  007762  001041                          BNE     2$              ;BR IF YES
2520
2521  007764  005037  012126          1$:     CLR     LOLIM
2522  007770  013737  003264  012130          MOV     MAXTRK,HILIM
2523  007776  004737  012024                  JSR     PC,RAND
2524  010002  010137  002546                  MOV     R1,DOTRK        ;GET RANDOM TRACK #
2525
2526  010006  005237  012126                  INC     LOLIM
2527  010012  012737  000032  012130          MOV     #32,HILIM
2528  010020  004737  012024                  JSR     PC,RAND
2529  010024  010137  002550                  MOV     R1,DOSEC        ;GET RANDOM SECTOR #
2530
2531  010030  004537  013010                  JSR     R5,CKOBAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
2532  010034  000753                             BR   1$              ;BR IF RETURN HERE
2533
2534  010036  012737  014622  014230          MOV     #DORSEC,DODISP  ;SET DISPATCH TABLE TO POINT TO
2535                                                                  ;READ SECTOR HANDLER.
2536  010044  005037  003262                  CLR     FIN             ;DO DX1 WHEN SET
2537  010050  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE.
2538
2539  010056  004737  012436                  JSR     PC,TIMDO
2540  010062  104117                          ERROR   117             ;DX0 NOT GETTING DONE FLAG
2541  010064  000240                          NOP
```

```
2542
2543  010066  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2544  010074  001041                          BNE     4$              ;BR IF YES
2545
2546  010076  005037  012126          3$:     CLR     LOLIM
2547  010102  013737  003264  012130          MOV     MAXTRK,HILIM
2548  010110  004737  012024                  JSR     PC,RAND
2549  010114  010137  003016                  MOV     R1,D1TRK        ;GET RANDOM TRACK #
2550
2551  010120  005237  012126                  INC     LOLIM
2552  010124  012737  000032  012130          MOV     #32,HILIM
2553  010132  004737  012024                  JSR     PC,RAND
2554  010136  010137  003020                  MOV     R1,D1SEC        ;GET RANDOM SECTOR #
2555
2556  010142  004537  013122                  JSR     R5,CK1BAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
2557  010146  000753                             BR   3$              ;BR IF RETURN HERE
2558
2559  010150  012737  016406  014236          MOV     #D1RSEC,D1DISP  ;SET DISPATCH TABLE TO POINT TO
2560                                                                  ;READ SECTOR HANDLER.
2561  010156  005037  003262                  CLR     FIN             ;GO BACK TO DX0 WHEN SET
2562  010162  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2563
2564  010170  004737  012532                  JSR     PC,TIMD1
2565  010174  104120                          ERROR   120             ;DX1 NOT GETTING DONE FLAG
2566  010176  000240                          NOP
2567
2568  010200  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2569  010206  001004                          BNE     5$              ;BR IF YES
2570  010210  023737  002542  003266          CMP     D0CNT,MAXSK     ;DID ALL SEEKS?
2571  010216  001262                          BNE     1$              ;BR IF NO
2572
2573  010220  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2574  010226  001004                          BNE     6$              ;BR IF YES
2575  010230  023737  003012  003266          CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
2576  010236  001317                          BNE     3$              ;BR IF NO
2577
2578  010240  005737  003274          6$:     TST     COMP2           ;DOING COMPAT PASS 2?
2579  010244  001402                          BEQ     DXTST3          ;BR IF NO
2580  010246  000137  011176                  JMP     EOP             ;ELSE ALL DONE
2581
```

```
                                ;;*****************************************************************
                                ;       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.
                                ;;*****************************************************************

010252  032737  000400  001246  DXTST3: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
010260  001062                        BNE     DXTST4          ;BR IF YES

010262  012737  000003  003260          MOV     #3,DXTST        ;3 = INIT TEST
010270  005037  002544                  CLR     DOERR           ;FLAGS
010274  005037  002554                  CLR     DOCERR

010300  012737  015666  014230          MOV     #DOINIT,DODISP  ;SET DISPATCH TABLE TO GO TO INIT HANDLER

010306  005037  003262                  CLR     FIN
010312  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

010320  004737  012436                  JSR     PC,TIMDO
010324  104046                          ERROR   46              ;DX0 HUNG ON INITIALIZE COMMAND
010326  000437                          BR      DXTST4

010330  005737  177170                  TST     RXCS            ;ERROR?
010334  100001                          BPL     1$              ;BR IF NO
010336  104063                          ERROR   63              ;INIT COMMAND ERROR

010340  032737  000001  177172  1$:     BIT     #ID,RXES        ;INIT DONE BIT SET?
010346  001001                          BNE     2$              ;BR IF YES
010350  104055                          ERROR   55              ;INIT DONE NOT SET

010352  005037  002544          2$:     CLR     DOERR
010356  005037  002554                  CLR     DOCERR
010362  012737  000001  002546          MOV     #1,DOTRK        ;EXP TRK & SEC
010370  012737  000001  002550          MOV     #1,DOSEC

010376  012737  015102  014230          MOV     #DOEBUF,DODISP  ;SETUP DISPATCH TABLE TO
                                                                ;EMPTY BUFFER & VERIFY DATA
010404  005037  003262                  CLR     FIN
010410  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

010416  004737  012436                  JSR     PC,TIMDO
010422  104056                          ERROR   56              ;DX0 HUNG ON EMPTY BUFF COMMAND
010424  000240                          NOP
```

```
                                        ;;*****************************************************************
                                        ;       RESTORE COMMAND TEST
                                        ;
                                        ;THIS TEST VERIFIES THAT THE RESTORE COMMAND CAN BE ISSUED
                                        ;TO DX0 & DX1 & THAT NO ERRORS RESULT FROM IT.
                                        ;;*****************************************************************

010426  032737  000400  001246  DXTST4: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
010434  001121                          BNE     DXTST5          ;BR IF YES

010436  012737  000004  003260          MOV     #4,DXTST        ;4 = RESTORE TEST
010444  012737  015712  014230          MOV     #DOREST,DODISP  ;SETUP DISPATCH TABLE
010452  005037  003262                  CLR     FIN
010456  052737  000100  177170          BIS     #IE,RXCS

010464  004737  012436                  JSR     PC,TIMDO
010470  104047                          ERROR   47              ;DX0 HUNG ON RESTORE COMMAND
010472  000431                          BR      2$

010474  005737  177170                  TST     RXCS            ;ERROR?
010500  100001                          BPL     1$              ;BR IF NO
010502  104054                          ERROR   54              ;DX0 RESTORE COMMAND ERROR

010504  005037  002546          1$:     CLR     DOTRK           ;CHK FOR RECORD NOT FOUND (RNF)
010510  012737  000001  002550          MOV     #1,DOSEC        ;SET TRK/SEC. SHOULD NOT MOVE
                                                                ;LOGIC THINKS ITS ALREADY THERE.
010516  005037  002544                  CLR     DOERR
010522  005037  002554                  CLR     DOCERR
010526  005037  003262                  CLR     FIN
010532  012737  014622  014230          MOV     #DORSEC,DODISP  ;SETUP TO READ SECTOR
010540  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERR LOOSE

010546  004737  012436                  JSR     PC,TIMDO
010552  104121                          ERROR   121             ;DX0 HUNG ON READ CMD
010554  000240                          NOP

010556  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
010564  001045                          BNE     DXTST5          ;BR IF YES

010566  012737  015730  014236          MOV     #D1REST,D1DISP  ;REPEAT FOR DX1
010574  005037  003262                  CLR     FIN
010600  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE

010606  004737  012532                  JSR     PC,TIMD1
010612  104122                          ERROR   122             ;DX1 HUNG ON RESTORE CMD
010614  000431                          BR      DXTST5

010616  005737  177170                  TST     RXCS            ;ERROR?
010622  100001                          BPL     3$              ;BR IF NO
010624  104074                          ERROR   74              ;DX1 RESTORE CMD ERROR
```

```
2680
2681  010626  005037  003016          3$:     CLR     D1TRK           ;READ TO CHECK FOR RNF
2682  010632  012737  000001  003020          MOV     #1,D1SEC
2683  010640  005037  003014                  CLR     D1ERR
2684  010644  005037  003024                  CLR     D1CERR
2685  010650  005037  003262                  CLR     FIN
2686  010654  012737  016406  014236          MOV     #D1RSEC,D1DISP  ;SETUP TO READ SECTOR
2687  010662  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE
2688
2689  010670  004737  012532                  JSR     PC,TIMD1
2690  010674  104123                          ERROR   123             ;DX1 HUNG ON READ CMD
2691  010676  000240                          NOP
2692
2693
2694
2695                                  ;;**********************************************************
2696                                  ;       WRITE SECTOR WITH DELETED DATA MARK TEST
2697                                  ;
2698                                  ;THIS TEST VERIFIES THAT THE WRITE DELETED DATA COMMAND CAN BE ISSUED
2699                                  ;& THAT THE 'DELETED DATA' BIT 5 IS SET IN RXCS AFTER READY IS RECV'D.
2700                                  ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2701                                  ;;**********************************************************
2702
2703  010700  032737  000400  001246  DXTST5: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2704  010706  001043                          BNE     DXTST6          ;BR IF YES
2705
2706  010710  012737  000005  003260          MOV     #5,DXTST        ;5 = WR DEL DATA TEST
2707  010716  005037  002544                  CLR     D0ERR
2708  010722  005037  002554                  CLR     D0CERR
2709
2710  010726  012737  014246  014230          MOV     #D0FBUF,D0DISP  ;SETUP DISPATCH TABLE
2711
2712  010734  005037  003262                  CLR     FIN
2713  010740  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2714
2715  010746  004737  012436                  JSR     PC,TIMD0
2716  010752  104050                          ERROR   50              ;DX0 HUNG ON WRITE DEL DATA CMD
2717  010754  000420                          BR      DXTST6
2718
2719  010756  012737  015746  014230          MOV     #D0STAT,D0DISP  ;SETUP DISPATCH TABLE
2720
2721  010764  005037  003262                  CLR     FIN
2722  010770  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2723
2724  010776  004737  012436                  JSR     PC,TIMD0
2725  011002  104051                          ERROR   51              ;DX0 HUNG ON READ STATUS
2726  011004  000404                          BR      DXTST6
2727
2728  011006  005737  177170                  TST     RXCS            ;ERROR?
2729  011012  100001                          BPL     .+4             ;BR IF NO
2730  011014  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).
                                ;;*****************************************************************

011016  032737  000400  001246  DXTST6: BIT     #BIT8,$DEVM     ;DROP DXC TESTS?
011024  001062                          BNE     4$              ;BR IF YES

011026  005037  011174                  CLR     INVCT
011032  012737  000006  003260          MOV     #6,DXTST        ;6 = INV ADDR TESTS
011040  012737  000115  002546          MOV     #115,DOTRK
011046  012737  000001  002550          MOV     #1,DOSEC
011054  012737  046401  177174          MOV     #46401,RXSA     ;INVALID TRACK

011062  012737  015764  014230  1$:     MOV     #DOINV,DODISP   ;SETUP DISPATCH TABLE

011070  005037  003262                  CLR     FIN
011074  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE

011102  004737  012436                  JSR     PC,TIMDO
011106  104060                          ERROR   60              ;DXO HUNG ON INVALID ADDR
011110  000432                          BR      EOP

011112  032737  000002  177172          BIT     #INVADR,RXES    ;INV ADDR BIT SET?
011120  001002                          BNE     2$              ;BR IF YES
011122  104061                          ERROR   61              ;INVALID ADDR BIT NOT SET
011124  000404                          BR      3$

011126  005737  177170          2$:     TST     RXCS            ;ERROR BIT SET?
011132  100401                          BMI     3$              ;BR IF YES
011134  104062                          ERROR   62              ;ERROR BIT NOT SET

011136  005737  011174          3$:     TST     INVCT           ;DID WE JUST DO INV SECTOR?
011142  001013                          BNE     4$              ;BR IF YES...DONE

011144  005237  011174                  INC     INVCT
011150  005037  002546                  CLR     DOTRK
011154  012737  000033  002550          MOV     #33,DOSEC
011162  012737  000033  177174          MOV     #33,RXSA        ;INVALID SECTOR
011170  000734                          BR      1$              ;REPEAT FOR INVALID SECTOR

011172  000401                  4$:     BR      .+4

011174  000000                  INVCT:  0                       ;0 = DOING INVALID TRACK
                                                                ;1 = DOING INVALID SECTOR
```

```
2781                                    ;;*****************************************************************
2782                                    ;       END OF PASS
2783                                    ;;*****************************************************************
2784
2785  011176  005237  001176      EOP:  INC     $PASS           ;PASS CTR
2786  011202  104401  021017            TYPE    ,ENDPAS
2787  011206  013746  001176            MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
2788  011212  104405                     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2789  011214  104401  021034            TYPE    ,MSG1
2790  011220  013746  027100            MOV     TERR,-(SP)      ;;SAVE TERR FOR TYPEOUT
2791  011224  104405                     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2792  011226  104401  021063            TYPE    ,MSG2
2793  011232  013746  027102            MOV     TSERR,-(SP)     ;;SAVE TSERR FOR TYPEOUT
2794  011236  104405                     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2795  011240  104401  021117            TYPE    ,MSG3
2796  011244  013746  027104            MOV     PERR,-(SP)      ;;SAVE PERR FOR TYPEOUT
2797  011250  104405                     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2798  011252  104401  021154            TYPE    ,MSG4
2799  011256  013746  027106            MOV     PSERR,-(SP)     ;;SAVE PSERR FOR TYPEOUT
2800  011262  104405                     TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2801  011264  104401  001334            TYPE    ,NULL           ;NULL CHAR
2802  011270  012737  000114  003264    MOV     #114,MAXTRK     ;FOR ALL SUBSEQUENT PASSES
2803  011276  012737  000764  003266    MOV     #500.,MAXSK
2804
2805  011304  005737  003272            TST     COMP1           ;COMPAT PASS 1?
2806  011310  001407                     BEQ     1$              ;BR IF NO
2807  011312  104401  021403            TYPE    ,COMPAT         ;DONE MSG
2808  011316  000000                     HALT
2809  011320  005037  003272            CLR     COMP1
2810  011324  005237  003274            INC     COMP2
2811  011330  000137  004072      1$:   JMP     LOOP            ;REPEAT
2812
2813  011334     377     377  000 NULL:  .BYTE   -1,-1,0         ;NULL CHAR STRING
2814          011340                     .EVEN
2815
```

```
                                ;;*****************************************************************
                                ;SYSTEM SIZER SUBROUTINE
                                ;;*****************************************************************

011340  012737  013176  000004  SIZE:   MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
011346  012737  000200  000006          MOV     #200,ERRVEC+2   ;LOCKOUT INTERR
011354  012737  013176  000010          MOV     #INTSRV,RESVEC  ;SETUP RESERVED INST TRAP ADDR
011362  012737  000200  000012          MOV     #200,RESVEC+2   ;LOCKOUT INTERR

                                ;;*****************************************************************
                                ;ROUTINE TO SIZE THE AMOUNT OF MEMORY
                                ;;*****************************************************************
011370  005037  013204                  CLR     INTFLG
011374  012700  003776                  MOV     #3776,R0        ;SIZE MEM. START WITH 1K
011400  005001                          CLR     R1              ;CTR, 1=1K, 2=2K, ETC
011402  005710          1$:     TST     (R0)
011404  005737  013204                  TST     INTFLG          ;GOT TIMEOUT INTERR?
011410  001007                          BNE     2$              ;BR IF YES
011412  005201                          INC     R1
011414  020027  167776                  CMP     R0,#167776      ;DONE 32K MEM?
011420  001405                          BEQ     3$              ;BR IF YES, ALL DONE
011422  062700  004000                  ADD     #4000,R0        ;GO TO NEXT 1K
011426  000765                          BR      1$

011430  162700  004000          2$:     SUB     #4000,R0        ;GO BACK TO LAST VALID 1K
011434  010037  012022          3$:     MOV     R0,MAXMEM       ;SAVE
011440  010146                          MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
                                                                ;;TYPE MEM SIZE
011442  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
011444  104401  020327                  TYPE    ,MEM

                                ;;*****************************************************************
                                ;ROUTINE TO CHECK IF "EIS-FIS" OPTION IS PRESENT
                                ;;*****************************************************************
011450  005037  012012                  CLR     EISPRES         ;CLEAR EIS-FIS PRESET FLAG
011454  005037  013204                  CLR     INTFLG          ;CLEAR TRAP FLAG
011460  072100                          ASH     R0,R1           ;EXECUTE EIS INSTRUCTION
011462  000240                          NOP
011464  005737  013204                  TST     INTFLG          ;TEST IF TRAP OCCURRED
011470  001003                          BNE     60$             ;BR IF YES
011472  005237  012012                  INC     EISPRES         ;NO TRAP - SET EIS-FIS PRESENT FLAG
011476  000404                          BR      61$             ;BR OVER TYPE-OUT
011500  104401  020307          60$:    TYPE    ,EISTXT         ;EIS-FIS TEXT
011504  104401  020350                  TYPE    ,NOTPRES
011510                          61$:
```

```
2862                                    ;;*****************************************************************
2863                                    ;ROUTINE TO CHECK IF CLUSTER PORT IS PRESENT
2864                                    ;;*****************************************************************
2865
2866  011510  005000                           CLR     R0
2867  011512  005037  013204                   CLR     INTFLG
2868  011516  012700  176500                   MOV     #TK1S,R0        ;SETUP CLUSTER CSR ADDR
2869  011522  005710                   4$:     TST     (R0)            ;DARE IT TO TIMEOUT
2870  011524  005737  013204                   TST     INTFLG          ;DID IT?
2871  011530  001011                           BNE     6$              ;BR IF YES
2872  011532  020027  176520                   CMP     R0,#TK3S        ;DID ALL 3?
2873  011536  001403                           BEQ     5$              ;BR IF YES
2874  011540  062700  000010                   ADD     #10,R0          ;ELSE DO NEXT
2875  011544  000766                           BR      4$
2876
2877  011546  005237  012014           5$:     INC     CLPRES          ;OPTION PRESENT
2878  011552  000406                           BR      7$              ;ALL 3 MUST RESPOND
2879
2880  011554  005037  012014           6$:     CLR     CLPRES          ;OPTION NOT PRESENT
2881  011560  104401  020136                   TYPE    ,CLOPT
2882  011564  104401  020350                   TYPE    ,NOTPRES
2883
2884  011570  012737  000006  000004   7$:     MOV     #ERRVEC+2,EPRVEC        ;RESET TIME-OUT VECTOR
2885  011576  005037  000006                   CLR     ERRVEC+2
2886  011602  012737  000012  000010           MOV     #RESVEC+2,RESVEC        ;RESET RESERVED INST VECTOR
2887  011610  005037  000012                   CLR     RESVEC+2
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO TYPEOUT DROPPED ITEMS
                                ;;*****************************************************************
011614  005000                          CLR     R0
011616  012701  000100                  MOV     #BIT6,R1
                                
011622  030137  001246        8$:       BIT     R1,$DEVM        ;DEVICE DROPPED?
011626  001006                          BNE     10$             ;BR IF YES
011630  000241                9$:       CLC
011632  006301                          ASL     R1
011634  103425                          BCS     13$             ;BR IF ALL BITS TESTED
011636  062700  000002                  ADD     #2,R0           ;ELSE BUMP INDEX
011642  000767                          BR      8$
                                
011644  016037  011664  011654 10$:     MOV     12$(R0),11$
011652  104401                          TYPE
011654  000000                11$:      .WORD   0               ;DEVICE TO BE DROPPED
011656  104401  020663                  TYPE    ,DROP
011662  000762                          BR      9$
                                
011664  020307                12$:      EISTXT
011666  020300                          CLOCK
011670  020266                          DRV0
011672  020273                          DRV1
011674  020211                          SCOM
011676  020224                          ACOM
011700  020156                          CLT1
011702  020167                          CLT2
011704  020200                          CLT3
011706  020240                          PRINT
                                ;;*****************************************************************
                                ;ROUTINE TO CHECK IF PRINTER EXISTS
                                ;;*****************************************************************
                                
011710  005037  012016        13$:      CLR     PRPRES
011714  005037  012020                  CLR     PRPORT
011720  005737  177514                  TST     PRS             ;SEE IF PRINTER ERR
011724  100027                          BPL     16$             ;BR IF NO
                                
011726  032737  000010  001246          BIT     #BIT3,$DEVM     ;COMM PORT IN EXT LOOPBACK?
011734  001405                          BEQ     15$             ;BR IF YES
                                
011736  104401  020240        14$:      TYPE    ,PRINT
011742  104401  020350                  TYPE    ,NOTPRES
011746  000420                          BR      17$
                                
011750  012737  037034  177420 15$:     MOV     #<AMOD!B9600!CPAR!CHAR8>,PARAM
011756  042737  000006  176610          BIC     #<RTS!DTR>,AMRC ;MUST BE TOGGLED
011764  052737  000006  176610          BIS     #<RTS!DTR>,AMRC
011772  005737  177514                  TST     PRS             ;NOW ANY ERRORS?
011776  100757                          BMI     14$             ;BR IF YES
012000  005237  012020                  INC     PRPORT
012004  005237  012016        16$:      INC     PRPRES          ;PRINTER PRESENT
                                
012010  000207                17$:      RTS     PC
```

```
2944
2945  012012  000000                  EIPRES: 0                       ;EIS-FIS PRESENT = NON-ZERO
2946  012014  000000                  CLPRES: 0                       ;CLUSTER PRESENT = NON-ZERO
2947  012016  000000                  PRPRES: 0                       ;PRINTER PRESENT = NON-ZERO
2948  012020  000000                  PRPORT: 0                       ;1 = PRINTER ENABLED THRU LOOPBACK
2949
2950  012022  000000                  MAXMEM: 0                       ;MAX MEMORY
2951
2952                                  ;;*****************************************************************
2953                                  ;       RANDOM NUMBER GENERATOR
2954                                  ;;*****************************************************************
2955
2956  012024  013700  012134  RAND:   MOV     LONUM,R0
2957  012030  013701  012132          MOV     HINUM,R1
2958  012034  012702  177771          MOV     #-7,R2
2959  012040  006300          1$:     ASL     R0
2960  012042  006101                  ROL     R1              ;ROTATE CARRY TO R1
2961  012044  005202                  INC     R2              ;DONE?
2962  012046  001374                  BNE     1$              ;BR IN NO
2963  012050  063700  012134          ADD     LONUM,R0        ;ADD NUMBER TO MAKE X 129
2964  012054  005501                  ADC     R1
2965  012056  063701  012132          ADD     HINUM,R1        ;ADD NUMBER TO MAKE X 129
2966  012062  062700  001057          ADD     #1057,R0        ;ADD LO CONSTANT
2967  012066  005501                  ADC     R1
2968  012070  062701  047401          ADD     #47401,R1       ;ADD HI CONSTANT
2969  012074  010037  012134          MOV     R0,LONUM
2970  012100  010137  012132          MOV     R1,HINUM
2971                                                          ;SCALE TO BE WITHIN LIMITS
2972  012104  042701  177400          BIC     #177400,R1      ;SAVE LO BYTE ONLY
2973  012110  020137  012126          CMP     R1,LOLIM
2974  012114  103743                  BLO     RAND            ;BR IF N < LOLIM
2975  012116  020137  012130          CMP     R1,HILIM
2976  012122  101340                  BHI     RAND            ;BR IF N > HILIM
2977  012124  000207                  RTS     PC              ;ELSE EXIT WITH R1 = N
2978
2979  012126  000000          LOLIM:  0
2980  012130  000000          HILIM:  0
2981
2982  012132  176543          HINUM:  .WORD 176543
2983  012134  123456          LONUM:  .WORD 123456
2984
2985
2986  012136  123727  001210  000001  EXAMKB: CMPB    $ENV,#1         ;APT?
2987  012144  001401                  BEQ     1$              ;BR IF YES
2988  012146  104407                  CKSWR
2989  012150  000207          1$:     RTS     PC
```

```
2990
2991                                    ;;*****************************************************************
2992                                    ;ROUTINE TO DISPLAY CURRENT DEVM & ALLOW OPERATOR TO INPUT NEW VALUE.
2993                                    ;;*****************************************************************
2994
2995  012152  104401  021500    GT$DEV: TYPE    ,DEVM           ;ELSE SHOW CURRENT
2996  012156  013746  001246            MOV     $DEVM,-(SP)     ;;SAVE $DEVM FOR TYPEOUT
2997  012162  104402                      TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2998  012164  104401  026250            TYPE    ,$MNEW          ;GET NEW
2999
3000  012170  005046            1$:     CLR     -(SP)           ;CLR CTR
3001  012172  005046                    CLR     -(SP)           ;CLR NEW DEVM
3002
3003  012174  105777  166744    2$:     TSTB    @$TKS           ;CHAR THERE?
3004  012200  100375                    BPL     2$              ;BR IF NO
3005  012202  117746  166740            MOVB    @$TKB,-(SP)     ;ELSE GET CHAR
3006  012206  042716  177600            BIC     #^C177,(SP)     ;MAKE IT 7 BIT ASCII
3007
3008  012212  021627  000015    3$:     CMP     (SP),#15        ;<CR>?
3009  012216  001013                    BNE     7$              ;BR IF NO
3010  012220  005766  000004            TST     4(SP)           ;ELSE IS IT 1'ST CHAR?
3011  012224  001403                    BEQ     4$              ;BR IF YES
3012  012226  016637  000002  001246    MOV     2(SP),$DEVM     ;ELSE SAVE NEW DEVM
3013  012234  062706  000006    4$:     ADD     #6,SP           ;RESTORE STACK
3014  012240  104401  001165    5$:     TYPE    ,$CRLF
3015  012244  000002            6$:     RTI
3016
3017  012246  004737  025350    7$:     JSR     PC,$TYPEC       ;ECHO CHAR
3018  012252  021627  000060            CMP     (SP),#60        ;CHAR < 0?
3019  012256  002420                    BLT     9$              ;BR IF YES
3020  012260  021627  000067            CMP     (SP),#67        ;CHAR >7?
3021  012264  003015                    BGT     9$              ;BR IF YES
3022  012266  042726  000060            BIC     #60,(SP)+       ;STRIP OFF ASCII
3023  012272  005766  000002            TST     2(SP)           ;IS IT 1'ST CHAR?
3024  012276  001403                    BEQ     8$              ;BR IF YES
3025  012300  006316                    ASL     (SP)            ;ELSE SHIFT PRESENT CHAR
3026  012302  006316                    ASL     (SP)            ;TO MAKE ROOM
3027  012304  006316                    ASL     (SP)            ;FOR NEW ONE
3028
3029  012306  005266  000002    8$:     INC     2(SP)           ;KEEP CHAR CT
3030  012312  056616  177776            BIS     -2(SP),(SP)     ;SET IN NEW CHAR
3031  012316  000726                    BR      2$              ;GET NEW ONE
3032
3033  012320  104401  001164    9$:     TYPE    ,$QUES          ;TYPE ?<CRLF>
3034  012324  000743                    BR      4$              ;DO ALL OVER
3035
```

```
3036                            ;;*****************************************************************
3037                            ;* ROUTINE TO SET 2 CONSECUTIVE INTERRUPT VECTORS
3038                            ;* 3 ARGUMENTS ARE PASSED THRU R5:
3039                            ;*      VECTOR ADDRESS
3040                            ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR RECVR
3041                            ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR XMITTER.
3042                            ;* PRIORITY WILL BE SET TO DISABLE FURTHUR INTERR.
3043                            ;;*****************************************************************
3044
3045  012326  010046            SETVEC: MOV     R0,-(SP)                ;SAVE
3046  012330  012500                    MOV     (R5)+,R0                ;GET VECTOR ADDR
3047  012332  012520                    MOV     (R5)+,(R0)+             ;PUT SERV ROUTINE ADDR THERE
3048  012334  012720  000200            MOV     #200,(R0)+              ;DISABLE INTERR
3049  012340  012520                    MOV     (R5)+,(R0)+             ;GET XMIT SERV ADDR NEXT
3050  012342  012710  000200            MOV     #200,(R0)
3051  012346  012600                    MOV     (SP)+,R0                ;RESTORE
3052  012350  000205                    RTS     R5
3053
3054
3055
3056
3057
3058                            ;;*****************************************************************
3059                            ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
3060                            ;RETURN    IF TIMED OUT
3061                            ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3062                            ;;*****************************************************************
3063
3064  012352  012537  012432    TIMER:  MOV     (R5)+,FLGHLD    ;GET FLAG
3065  012356  012537  012434            MOV     (R5)+,CTHLD     ;GET COUNT THAT FLAG SHOULD GO TO
3066
3067  012362  012701  000010            MOV     #10,R1
3068  012366  012700  177777    4$:     MOV     #-1,R0
3069  012372  027737  000034  012434 1$:CMP     @FLGHLD,CTHLD   ;RESPONDING?
3070  012400  001411                    BEQ     2$              ;BR IF YES
3071  012402  004737  012136            JSR     PC,EXAMKB
3072  012406  005300                    DEC     R0              ;ELSE DEC COUNTER
3073  012410  001370                    BNE     1$              ;BR IF NOT TIMED OUT
3074  012412  005237  001200            INC     $DEVCT          ;FOR APT
3075  012416  005301                    DEC     R1
3076  012420  001362                    BNE     4$
3077  012422  000402                    BR      .+6             ;ELSE TIMED OUT
3078
3079  012424  062705  000004    2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
3080  012430  000205                    RTS     R5
3081
3082  012432  000000            FLGHLD: 0                       ;FLAG
3083  012434  000000            CTHLD:  0                       ;COUNT THAT FLAG MUST REACH
3084
```

```
3085                                    ;;*****************************************************************
3086                                    ;WATCHDOG TIMER TO PREVENT DX0 FROM ENTERING AN ENDLESS WAIT LOOP
3087                                    ;RETURN     IF TIMED OUT
3088                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3089                                    ;;*****************************************************************
3090
3091  012436  005037  012530            TIMD0:  CLR     D0TMO
3092  012442  012737  000010  012526            MOV     #10,CTREG1
3093  012450  012737  177777  012524    4$:     MOV     #-1,CTREG
3094  012456  005737  003262            1$:     TST     FIN             ;FINISHED?
3095  012462  001015                            BNE     2$              ;BR IF YES
3096  012464  004737  012136                    JSR     PC,EXAMKB
3097  012470  005337  012524                    DEC     CTREG           ;ELSE DEC CTR
3098  012474  001370                            BNE     1$              ;BR IF NOT TIMED OUT
3099  012476  005237  001200                    INC     $DEVCT          ;FOR APT
3100  012502  005337  012526                    DEC     CTREG1
3101  012506  001360                            BNE     4$
3102  012510  005237  012530                    INC     D0TMO           ;SET FLAG
3103  012514  000402                            BR      .+6             ;TIMED OUT
3104
3105  012516  062716  000004            2$:     ADD     #4,(SP)         ;BUMP RET
3106  012522  000207                            RTS     PC
3107
3108  012524  000000                    CTREG:  0
3109  012526  000000                    CTREG1: 0
3110  012530  000000                    D0TMO:  0                       ;1 = TIMEOUT OCCURED
3111
3112
3113                                    ;;*****************************************************************
3114                                    ;WATCHDOG TIMER TO PREVENT DX1 FROM ENTERING AN ENDLESS WAIT LOOP
3115                                    ;RETURN     IF TIMED OUT
3116                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3117                                    ;;*****************************************************************
3118
3119  012532  005037  012620            TIMD1:  CLR     D1TMO
3120  012536  012737  000010  012526            MOV     #10,CTREG1
3121  012544  012737  177777  012524    4$:     MOV     #-1,CTREG
3122  012552  005737  003262            1$:     TST     FIN             ;FINISHED?
3123  012556  001015                            BNE     2$              ;BR IF YES
3124  012560  004737  012136                    JSR     PC,EXAMKB
3125  012564  005337  012524                    DEC     CTREG           ;ELSE DEC CTR
3126  012570  001370                            BNE     1$              ;BR IF NOT TIMED OUT
3127  012572  005237  001200                    INC     $DEVCT          ;FOR APT
3128  012576  005337  012526                    DEC     CTREG1
3129  012602  001360                            BNE     4$
3130  012604  005237  012620                    INC     D1TMO           ;SET FLAG
3131  012610  000402                            BR      .+6             ;TIMED OUT
3132
3133  012612  062716  000004            2$:     ADD     #4,(SP)         ;BUMP RET
3134  012616  000207                            RTS     PC
3135
3136  012620  000000                    D1TMO:  0                       ;1 = TIMEOUT OCCURED
3137
```

```
3138                                    ;;*****************************************************************
3139                                    ;       GENERAL TIMER
3140                                    ;;*****************************************************************
3141
3142  012622  012700  177777           TIMER1: MOV     #-1,R0
3143  012626  005300                           DEC     R0
3144  012630  001376                           BNE     .-2
3145  012632  000207                           RTS     PC
3146
3147                                    ;;*****************************************************************
3148                                    ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
3149                                    ;RETURN    IF TIMED OUT
3150                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3151                                    ;;*****************************************************************
3152
3153  012634  012537  012714           TIMER2: MOV     (R5)+,REGHLD    ;GET REG
3154  012640  012537  012716                   MOV     (R5)+,BITHLD    ;GET BIT TO BE TESTED
3155
3156  012644  012701  000010                   MOV     #10,R1
3157  012650  012700  177777           4$:     MOV     #-1,R0
3158  012654  033777  012716  000032   1$:     BIT     BITHLD,@REGHLD  ;BIT THERE?
3159  012662  001011                           BNE     2$              ;BR IF YES
3160  012664  004737  012136                   JSR     PC,EXAMKB
3161  012670  005300                           DEC     R0              ;ELSE DEC COUNTER
3162  012672  001370                           BNE     1$              ;BR IF NOT TIMED OUT
3163  012674  005237  001200                   INC     $DEVCT          ;FOR APT
3164  012700  005301                           DEC     R1
3165  012702  001362                           BNE     4$
3166  012704  000402                           BR      .+6             ;ELSE TIMED OUT
3167
3168  012706  062705  000004           2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
3169  012712  000205                           RTS     R5
3170
3171  012714  000000                   REGHLD: 0                       ;DEVICE REG
3172  012716  000000                   BITHLD: 0                       ;DEV REG BIT TO BE TESTED
3173
3174
3175                                    ;;*****************************************************************
3176                                    ;ROUTINE TO CLEAR BOTH BAD TRACK/SECTOR TABLES
3177                                    ;;*****************************************************************
3178
3179  012720  012700  002764           CLRBAD: MOV     #D0BAD,R0
3180  012724  005020                   1$:     CLR     (R0)+
3181  012726  020027  003010                   CMP     R0,#D0BAD+20.
3182  012732  001374                           BNE     1$
3183
3184  012734  012700  003234                   MOV     #D1BAD,R0
3185  012740  005020                   2$:     CLR     (R0)+
3186  012742  020027  003260                   CMP     R0,#D1BAD+20.
3187  012746  001374                           BNE     2$
3188
3189  012750  000207                           RTS     PC
3190
```

```
3191                                    ;;*****************************************************************
3192                                    ;LOAD DX0 BAD TRACK/SECTOR TABLE
3193                                    ;;*****************************************************************
3194
3195  012752  010046                    LDOBAD: MOV     R0,-(SP)        ;SAVE
3196
3197  012754  012700  002764                    MOV     #DOBAD,R0
3198  012760  005710                    1$:     TST     (R0)            ;ENTRY PRESENT?
3199  012762  001003                            BNE     2$              ;BR IF YES
3200  012764  013710  177174                    MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
3201  012770  000405                            BR      3$              ;EXIT
3202
3203  012772  005720                    2$:     TST     (R0)+           ;BUMP PTR
3204  012774  020027  003010                    CMP     R0,#DOBAD+20.   ;AT END OF TABLE?
3205  013000  001367                            BNE     1$              ;BR IF NO
3206  013002  104104                            ERROR   104             ;10 BAD SECTORS...REPLACE
3207  013004  012600                    3$:     MOV     (SP)+,R0        ;RESTORE
3208  013006  000207                            RTS     PC
3209
3210
3211
3212
3213
3214                                    ;;*****************************************************************
3215                                    ;CHECK DX0 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
3216                                    ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
3217                                    ;;*****************************************************************
3218
3219  013010  010046                    CKOBAD: MOV     R0,-(SP)        ;SAVE
3220  013012  010146                            MOV     R1,-(SP)
3221
3222  013014  013701  002546                    MOV     DOTRK,R1
3223  013020  000301                            SWAB    R1
3224  013022  053701  002550                    BIS     DOSEC,R1
3225
3226  013026  012700  002764                    MOV     #DOBAD,R0
3227  013032  020027  003010            1$:     CMP     R0,#DOBAD+20.   ;END OF TABLE?
3228  013036  001405                            BEQ     2$              ;BR IF YES
3229  013040  005710                            TST     (R0)            ;ELSE ANY ENTRY?
3230  013042  001403                            BEQ     2$              ;BR IF NO
3231  013044  022001                            CMP     (R0)+,R1        ;ELSE COMPARE?
3232  013046  001371                            BNE     1$              ;BR IF NO
3233  013050  000402                            BP      3$              ;ELSE DONT BUMP RET ADDR
3234
3235  013052  062705  000002            2$:     ADD     #2,R5           ;BUMP RET ADDR
3236  013056  012601                    3$:     MOV     (SP)+,R1        ;RESTORE
3237  013060  012600                            MOV     (SP)+,R0
3238  013062  000205                            RTS     R5
```

```
3239                            ;;*****************************************************************
3240                            ;LOAD DX1 BAD TRACK/SECTOR TABLE
3241                            ;;*****************************************************************
3242
3243  013064  010046            LD1BAD: MOV     R0,-(SP)        ;SAVE
3244
3245  013066  012700  003234            MOV     #D1BAD,R0
3246  013072  005710            1$:     TST     (R0)            ;ENTRY PRESENT?
3247  013074  001003                    BNE     2$              ;BR IF YES
3248  013076  013710  177174            MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
3249  013102  000405                    BR      3$              ;EXIT
3250
3251  013104  005720            2$:     TST     (R0)+           ;BUMP PTR
3252  013106  020027  003260            CMP     R0,#D1BAD+20.   ;AT END OF TABLE?
3253  013112  001367                    BNE     1$              ;BR IF NO
3254  013114  104105                    ERROR   105             ;10 BAD SECTORS...REPLACE
3255  013116  012600            3$:     MOV     (SP)+,R0        ;RESTORE
3256  013120  000207                    RTS     PC
3257
3258
3259
3260
3261
3262                            ;;*****************************************************************
3263                            ;CHECK DX1 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
3264                            ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
3265                            ;;*****************************************************************
3266
3267  013122  010046            CK1BAD: MOV     R0,-(SP)        ;SAVE
3268  013124  010146                    MOV     R1,-(SP)
3269
3270  013126  013701  003016            MOV     D1TRK,R1
3271  013132  000301                    SWAB    R1
3272  013134  053701  003020            BIS     D1SEC,R1
3273
3274  013140  012700  003234            MOV     #D1BAD,R0
3275  013144  020027  003260    1$:     CMP     R0,#D1BAD+20.   ,END OF TABLE?
3276  013150  001405                    BEQ     2$              ;BR IF YES
3277  013152  005710                    TST     (R0)            ;ELSE ANY ENTRY?
3278  013154  001403                    BEQ     2$              ;BR IF NO
3279  013156  022001                    CMP     (R0)+,R1        ;ELSE COMPARE?
3280  013160  001371                    BNE     1$              ;BR IF NO
3281  013162  000402                    BP      3$              ;ELSE DONT BUMP RET ADDR
3282
3283  013164  062705  000002    2$:     ADD     #2,R5           ;BUMP RET ADDR
3284  013170  012601            3$:     MOV     (SP)+,R1        ;RESTORE
3285  013172  012600                    MOV     (SP)+,R0
3286  013174  000205                    RTS     R5
3287
```

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

013176  005237  013204        INTSRV: INC     INTFLG
013202  000002                        RTI
013204  000000                INTFLG: 0


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

013206  013737  177514  013336  PRSRV:  MOV     PRS,SPRS                ;STORE
013214  005737  013336                  TST     SPRS            ;ERROR?
013220  100002                        BPL     1$              ;BR IF NO
013222  104007                        ERROR   7               ;PRINTER STATUS ERROR
013224  000002                        RTI

013226  013737  013332  177516  1$:     MOV     PRCHR,PRB       ;ELSE PRINT CHAR
013234  023727  013332  000015          CMP     PRCHR,#CR       ;JUST DID CR?
013242  001413                        BEQ     2$              ;BR IF YES
013244  023727  013332  000012          CMP     PRCHR,#LF       ;JUST DID LF?
013252  001413                        BEQ     3$              ;BR IF YES
013254  023727  013332  000176          CMP     PRCHR,#176      ;JUST DID LAST CHAR?
013262  001413                        BEQ     4$              ;BR IF YES
013264  005237  013332                  INC     PRCHR           ;ELSE BUMP CHAR FOR NEXT INTERRUPT
013270  000417                        BR      5$              ;EXIT

013272  012737  000012  013332  2$:     MOV     #LF,PRCHR       ;DO LF NEXT
013300  000413                        BR      5$
013302  012737  000040  013332  3$:     MOV     #40,PRCHR       ;DO SPACE NEXT
013310  000407                        BR      5$
013312  012737  000015  013332  4$:     MOV     #CR,PRCHR       ;DO CR NEXT & START OVER
013320  005237  013334                  INC     LINCT
013324  005237  001200                  INC     $DEVCT          ;FOR APT
013330  000002                5$:     RTI

013332  000000                PRCHR:  0                       ;CHAR TO BE PRINTED
013334  000000                LINCT:  0                       ;LINE CTR
013336  000000                SPRS:   0                       ;STORE PRINTER CSR
```

```
3331
3332                                    ;;*****************************************************************
3333                                    ;CLUSTER TERMINAL #1 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3334                                    ;;*****************************************************************
3335
3336  013340  042737  000100  176504  TP1SRV: BIC     #IE,TP1S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3337  013346  013737  013434  176506          MOV     T1CHR,TP1B      ;PRINT CHAR
3338  013354  000002                          RTI
3339
3340  013356  013737  176502  013436  TK1SRV: MOV     TK1B,T1HLD      ;STORE CHAR
3341  013364  023737  013436  013434          CMP     T1HLD,T1CHR     ;OK?
3342  013372  001401                          BEQ     1$              ;BR IF YES
3343  013374  104010                          ERROR   10              ;CHAR COMP ERROR
3344
3345  013376  023727  013434  000377  1$:     CMP     T1CHR,#377      ;LAST CHAR?
3346  013404  001403                          BEQ     2$              ;BR IF YES
3347  013406  005237  013434                  INC     T1CHR           ;ELSE BUMP
3348  013412  000404                          BR      3$
3349
3350  013414  005037  013434          2$:     CLR     T1CHR           ;START OVER
3351  013420  005237  001200                  INC     $DEVCT          ;FOR APT
3352  013424  052737  000100  176504  3$:     BIS     #IE,TP1S        ;ALLOW PRINTER INTERR
3353  013432  000002                          RTI
3354
3355  013434  000000                  T1CHR:  0                       ;CHAR TO XMITT
3356  013436  000000                  T1HLD:  0                       ;REC'D CHAR
3357
3358
3359                                    ;;*****************************************************************
3360                                    ;CLUSTER TERMINAL #2 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3361                                    ;;*****************************************************************
3362
3363  013440  042737  000100  176514  TP2SRV: BIC     #IE,TP2S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3364  013446  013737  013534  176516          MOV     T2CHR,TP2B      ;PRINT CHAR
3365  013454  000002                          RTI
3366
3367  013456  013737  176512  013536  TK2SRV: MOV     TK2B,T2HLD      ;STORE CHAR
3368  013464  023737  013536  013534          CMP     T2HLD,T2CHR     ;OK?
3369  013472  001401                          BEQ     1$              ;BR IF YES
3370  013474  104011                          ERROR   11              ;CHAR COMP ERROR
3371
3372  013476  023727  013534  000377  1$:     CMP     T2CHR,#377      ;LAST CHAR?
3373  013504  001403                          BEQ     2$              ;BR IF YES
3374  013506  005237  013534                  INC     T2CHR           ;ELSE BUMP
3375  013512  000404                          BR      3$
3376
3377  013514  005037  013534          2$:     CLR     T2CHR           ;START OVER
3378  013520  005237  001200                  INC     $DEVCT          ;FOR APT
3379  013524  052737  000100  176514  3$:     BIS     #IE,TP2S        ;ALLOW PRINTER INTERR
3380  013532  000002                          RTI
3381
3382  013534  000000                  T2CHR:  0                       ;CHAR TO XMITT
3383  013536  000000                  T2HLD:  0                       ;REC'D CHAR
```

```
3384
3385                                    ;;*****************************************************************
3386                                    ;CLUSTER TERMINAL #3 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3387                                    ;;*****************************************************************
3388
3389  013540  042737  000100  176524  TP3SRV: BIC     #IE,TP3S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3390  013546  013737  013634  176526          MOV     T3CHR,TP3B      ;PRINT CHAR
3391  013554  000002                          RTI
3392
3393  013556  013737  176522  013636  TK3SRV:: MOV     TK3B,T3HLD      ;STORE CHAR
3394  013564  023737  013636  013634          CMP     T3HLD,T3CHR     ;OK?
3395  013572  001401                          BEQ     1$              ;BR IF YES
3396  013574  104012                          ERROR   12              ;CHAR COMP ERROR
3397
3398  013576  023727  013634  000377  1$:     CMP     T3CHR,#377      ;LAST CHAR?
3399  013604  001403                          BEQ     2$              ;BR IF YES
3400  013606  005237  013634                  INC     T3CHR           ;ELSE BUMP
3401  013612  000404                          BR      3$
3402
3403  013614  005037  013634          2$:     CLR     T3CHR           ;START OVER
3404  013620  005237  001200                  INC     $DEVCT          ;FOR APT
3405  013624  052737  000100  176524  3$:     BIS     #IE,TP3S        ;ALLOW PRINTER INTERR
3406  013632  000002                          RTI
3407
3408  013634  000000                  T3CHR:  0                       ;CHAR TO XMITT
3409  013636  000000                  T3HLD:  0                       ;REC'D CHAR
3410
3411
3412                                    ;;*****************************************************************
3413                                    ;ASYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
3414                                    ;;*****************************************************************
3415
3416  013640  042737  000100  176614  AMXSRV: BIC     #IE,AMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3417  013646  013737  013734  176616          MOV     COMCHR,AMXB     ;XMITT CHAR
3418  013654  000002                          RTI
3419
3420  013656  013737  176612  013736  AMRSRV: MOV     AMRB,COMHLD     ;STORE CHAR
3421  013664  023737  013736  013734          CMP     COMHLD,COMCHR   ;OK?
3422  013672  001401                          BEQ     1$              ;BR IF YES
3423  013674  104013                          ERROR   13              ;CHAR COMPARE ERROR
3424
3425  013676  023727  013734  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
3426  013704  001403                          BEQ     2$              ;BR IF YES
3427  013706  005237  013734                  INC     COMCHR          ;ELSE BUMP
3428  013712  000404                          BR      3$
3429
3430  013714  005037  013734          2$:     CLR     COMCHR          ;START OVER
3431  013720  005237  001200                  INC     $DEVCT          ;FOR APT
3432  013724  052737  000100  176614  3$:     BIS     #IE,AMXC        ;ALLOW XMIT INTERR
3433  013732  000002                          RTI
3434
3435  013734  000000                  COMCHR: 0                       ;XMIT CHAR
3436  013736  000000                  COMHLD: 0                       ;REC'D CHAR
```

```
3437
3438                                    ;;*****************************************************************
3439                                    ;SYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 27 THRU 377.
3440                                    ;;*****************************************************************
3441
3442  013740  042737  000100  176624  SMXSRV: BIC     #IE,SMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3443  013746  013737  013734  176626          MOV     COMCHR,SMXB     ;XMITT CHAR
3444  013754  023727  013734  000377          CMP     COMCHR,#377     ;LAST CHAR?
3445  013762  001020                          BNE     2$              ;BR IF NO
3446
3447  013764  005737  014026                  TST     LSTCHR
3448  013770  001010                          BNE     1$
3449  013772  005237  014026                  INC     LSTCHR
3450  013776  052737  000100  176624          BIS     #IE,SMXC        ;ALLOW RDY INTR SO CAN SHUT DOWN FAST
3451  014004  005237  001200                  INC     $DEVCT          ;FOR APT
3452  014010  000002                          RTI
3453
3454  014012  042737  000020  176624  1$:     BIC     #SEND,SMXC      ;NO MORE TO SEND
3455  014020  005237  001200                  INC     $DEVCT          ;FOR APT
3456  014024  000002                  2$:     RTI
3457
3458  014026  000000                  LSTCHR: 0                       ;SET WHEN LAST CHAR XMIT
3459
3460
3461
3462
3463
3464  014030  013737  176622  013736  SMRSRV: MOV     SMRB,COMHLD     ;STORE CHAR
3465  014036  123727  013736  000026          CMPB    COMHLD,#026     ;IGNORE SYNC CHARS
3466  014044  001430                          BEQ     4$
3467  014046  023737  013736  013734          CMP     COMHLD,COMCHR   ;OK?
3468  014054  001401                          BEQ     1$              ;BR IF YES
3469  014056  104014                          ERROR   14              ;CHAR COMPARE ERROR
3470
3471  014060  023727  013734  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
3472  014066  001403                          BEQ     2$              ;BR IF YES
3473  014070  005237  013734                  INC     COMCHR          ;ELSE BUMP
3474  014074  000411                          BR      3$
3475
3476  014076  012737  177777  013734  2$:     MOV     #-1,COMCHR      ;INDICATE DONE
3477  014104  042737  000100  176620          BIC     #IE,SMRC        ;ALL DONE
3478  014112  005237  001200                  INC     $DEVCT          ;FOR APT
3479  014116  000403                          BR      4$
3480
3481  014120  052737  000100  176624  3$:     BIS     #IE,SMXC        ;ALLOW XMIT INTERR
3482  014126  000002                  4$:     RTI
3483
```

```
                                ;;*****************************************************************
                                ;                   DISK INTERRUPT HANDLERS
                                ;ALL DISK INTERRUPTS ENTER THRU RXSRV & DISPATCH TO THE
                                ;CURRENT SERVICE ROUTINE POINTED TO BY:
                                ;                   DODISP FOR DX0 INTERRUPTS
                                ;                   D1DISP FOR DX1 INTERRUPTS
                                ;WILL GENERALLY BE IN THE ORDER OF SERVICE ROUTINES BELOW.
                                ;;*****************************************************************

014130  105737  177170  RXSRV:  TSTB    RXCS            ;CONTR RDY?
014134  100413          1$:     BMI     1$              ;BR IF YES
014136  013737  177170  014240          MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
014144  013737  177172  014242          MOV     RXES,SRXES
014152  013737  177174  014244          MOV     RXSA,SRXSA
014160  104015                          ERROR   15              ;UNEXPECTED INTERRUPT
014162  000002                          RTI                     ;GO BACK

014164  005737  177170  1$:     TST     RXCS            ;ERROR?
014170  100011                          BPL     2$              ;BR IF NO

014172  013737  177170  014240          MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
014200  013737  177172  014242          MOV     RXES,SRXES
014206  013737  177174  014244          MOV     RXSA,SRXSA

014214  032737  000020  177170  2$:     BIT     #USEL,RXCS      ;CHECK UNIT SELECT BIT
014222  001003                          BNE     D1INT           ;BR IF INTER FROM DX1

014224  000177  000000                  JMP     @DODISP         ;ELSE DISPATCH TO DX0 SERVICE ROUTINE
014230  000000          DODISP: 0

014232  000177  000000  D1INT:  JMP     @D1DISP         ;DX1 INTERRUPT DISPATCH
014236  000000          D1DISP: 0


014240  000000          SRXCS:  0               ;SAVE RXCS
014242  000000          SRXES:  0               ;SAVE RXES
014244  000000          SRXSA:  0               ;SAVE RXSA
```

```
                                ;;*****************************************************************
                                ; HANDLER TO FILL THE DRIVE BUFFER FOR DX0
                                ;& POINT TO THE WRITE SECTOR HANDLER.
                                ;;*****************************************************************

014246  012703  000100          DOFBUF: MOV     #100,R3         ;WORD CT
014252  013701  002546                  MOV     DOTRK,R1
014256  000301                          SWAB    R1
014260  053701  002550                  BIS     DOSEC,R1
014264  012737  014312  014230          MOV     #DOWSEC,DODISP  ;POINT TO WRITE SECTOR AFTER RTI
014272  012737  000101  177170          MOV     #<FBUF!IE>,RXCS ;ISSUE FILL BUFFER CMD
014300  010137  177172          1$:     MOV     R1,RXDB         ;FILL ENTIRE SECTOR WITH ITS ADDRESS
014304  005303                          DEC     R3
014306  001374                          BNE     1$
014310  000002                          RTI


                                ;;*****************************************************************
                                ;HANDLER TO WRITE THE SECTOR FOR DX0 & POINT TO WRITE CHECK.
                                ;;*****************************************************************

014312  013701  002546          DOWSEC: MOV     DOTRK,R1
014316  000301                          SWAB    R1
014320  053701  002550                  BIS     DOSEC,R1
014324  010137  177174                  MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
014330  012737  014366  014230          MOV     #DOWCHK,DODISP  ;POINT TO WRITE CHECK HANDLER
014336  023727  003260  000005          CMP     DXTST,#5
014344  001004                          BNE     1$
014346  012737  000115  177170          MOV     #<WDDSEC!IE>,RXCS       ;ONLY FOR WRITE DELETED DATA TEST
014354  000403                          BR      2$
014356  012737  000105  177170  1$:     MOV     #<WSEC.IE>,RXCS ;ISSUE WRITE SECTOR COMMAND
014364  000002                  2$:     RTI
```

```
3560                                    ;;*****************************************************************
3561                                    ;HANDLER TO WRITE CHECK ON DX0.
3562                                    ;WILL GO TO FILL BUFFER TO DO SAME SECTOR ON SOFT ERROR
3563                                    ;OR NEXT SECTOR ON NO ERROR OR HARD ERROR.
3564                                    ;WILL GO TO READ SECTOR AFTER ALL SECTORS ON TRACK ARE WRITTEN.
3565                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3566                                    ;;*****************************************************************
3567
3568  014366  005737  177170       DOWCHK: TST     RXCS            ;ERROR?
3569  014372  100024                        BPL     1$              ;BR IN NO
3570  014374  023727  002544  000012        CMP     DOERR,#10.      ;10 ERRORS?
3571  014402  001072                        BNE     6$              ;BR IF NO & TRY AGAIN
3572  014404  004737  012752              JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
3573  014410  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
3574  014416  001410                        BEQ     7$              ;BR IF NO
3575  014420  104064                        ERROR   64              ;HARD SEEK ERROR WRITE SECTOR
3576  014422  012737  014500  014230        MOV     #2$,DODISP      ;GO TO 2$ AFTER RESTORE
3577  014430  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
3578  014436  000002                        RTI
3579
3580  014440  104016                7$:     ERROR   16              ;HARD ERROR WRITE SECTOR
3581  014442  000416                        BR      2$              ;GO TO NEXT SECTOR
3582
3583  014444  005737  002544        1$:     TST     DOERR           ;ANY PREV ERRORS?
3584  014450  001413                        BEQ     2$              ;BR IF NO
3585  014452  005237  027102                INC     TSERR           ;TOTAL SOFT ERR CT
3586  014456  005237  027106                INC     PSERR           ;PASS  SOFT ERR CT
3587  014462  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
3588  014470  001402                        BEQ     8$              ;BR IF NO
3589  014472  104065                        ERROR   65              ;SOFT SEEK ERROR WRITE SECTOR
3590  014474  000401                        BR      2$
3591  014476  104017                8$:     ERROR   17              ;SOFT ERROR WRITE SECTOR
3592
3593  014500  005037  002544        2$:     CLR     DOERR
3594  014504  023727  002550  000031        CMP     DOSEC,#31       ;LAST ODD SECTOR?
3595  014512  001004                        BNE     3$              ;BR IF NO
3596  014514  012737  000002  002550        MOV     #2,DOSEC        ;ELSE DO EVEN SECTORS NOW
3597  014522  000407                        BR      4$
3598  014524  023727  002550  000032  3$:   CMP     DOSEC,#32       ;LAST EVEN SECTOR?
3599  014532  001405                        BEQ     5$              ;BR IF YES
3600  014534  062737  000002  002550        ADD     #2,DOSEC        ;ELSE BUMP SECTOR
3601  014542  000137  014246        4$:     JMP     DOFBUF          ;GO TO FILL BUFFER & DO ANOTHER
3602
3603  014546  012737  000001  002550  5$:   MOV     #1,DOSEC        ;ALL SECTORS DONE...CHECK DATA
3604  014554  005037  002544                CLR     DOERR
3605  014560  005037  002552                CLR     DOCMER
3606  014564  000137  014622                JMP     DORSEC          ;GO TO READ SECTOR HANDLER
3607
3608  014570  005237  002544        6$:     INC     DOERR
3609  014574  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
3610  014602  001757                        BEQ     4$              ;BR IN NO
3611  014604  012737  014246  014230        MOV     #DOFBUF,DODISP  ;GOTO FILL BUFF AFTER RESTORE
3612  014612  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
3613  014620  000002                        RTI
3614
3615
```

```
3616                                    ;;*****************************************************************
3617                                    ;HANDLER TO READ A SECTOR ON DX0 & POINT TO CHECK.
3618                                    ;;*****************************************************************
3619
3620  014622  013701  002546            DORSEC: MOV     DOTRK,R1
3621  014626  000301                            SWAB    R1
3622  014630  053701  002550                    BIS     DOSEC,R1
3623  014634  010137  177174                    MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR
3624
3625  014640  012737  014656  014230            MOV     #DORCHK,DODISP  ;POINT TO READ CHECK HANDLER
3626  014646  012737  000107  177170            MOV     #<RSEC!IE>,RXCS ;ISSUE READ SECTOR COMMAND
3627  014654  000002                            RTI
3628
3629
3630                                    ;;*****************************************************************
3631                                    ;HANDLER TO READ CHECK ON DX0.
3632                                    ;WILL GO TO NEXT SECTOR/TRK IF HARD ERROR.
3633                                    ;WILL GO TO EMPTY BUFFER IF NO ERROR OR SOFT ERROR.
3634                                    ;WILL GO TO READ SECTOR (SAME) ON ERROR.
3635                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3636                                    ;;*****************************************************************
3637
3638  014656  005737  177170            DORCHK: TST     RXCS            ;ERROR?
3639  014662  100054                            BPL     1$              ;BR IF NO
3640  014664  023727  002544  000012            CMP     DOERR,#10.      ;10 ERRORS?
3641  014672  001415                            BEQ     3$              ;BR IF YES
3642  014674  005237  002544                    INC     DOERR           ;ELSE DO AGAIN
3643  014700  032737  000020  014242            BIT     #RNF,SRXES      ;SEEK ERROR?
3644  014706  001745                            BEQ     DORSEC          ;BR IF NO
3645  014710  012737  014622  014230            MOV     #DORSEC,DODISP  ;ELSE DO RESTORE & RET TO RD SEC
3646  014716  012737  000117  177170            MOV     #<RESTOR!IE>,RXCS
3647  014724  000002                            RTI
3648
3649  014726  004737  012752            3$:     JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
3650  014732  032737  000020  014242            BIT     #RNF,SRXES      ;SEEK ERROR?
3651  014740  001410                            BEQ     7$              ;BR IF NO
3652  014742  104066                            ERROR   66              ;SEEK ERROR READ SECTOR
3653
3654  014744  012737  015644  014230            MOV     #DONEXT,DODISP  ;DO RESTORE
3655  014752  012737  000117  177170            MOV     #<RESTOR!IE>,RXCS
3656  014760  000002                            RTI
3657
3658  014762  032737  000010  014242    7$:     BIT     #CRC,SRXES      ;CRC ERROR?
3659  014770  001404                            BEQ     10$             ;BR IF NO
3660  014772  005237  002562                    INC     DOCRC           ;ELSE SET FLAG
3661  014776  104106                            ERROR   106             ;HARD CRC ERROR
3662  015000  000423                            BR      2$              ;& GO TO EMP BUFF TO CHK DATA
3663
3664  015002  005037  002562            10$:    CLR     DOCRC
3665  015006  104020                            ERROR   20              ;HARD READ ERROR
3666  015010  000137  015644                    JMP     DONEXT
```

```
3667
3668  015014  005737  002544          1$:     TST     DOERR           ;ANY PREV ERRORS?
3669  015020  001413                          BEQ     2$              ;BR IF NO
3670  015022  005237  027102                  INC     TSERR           ;TOTAL SOFT ERR CT
3671  015026  005237  027106                  INC     PSERR           ;PASS  SOFT ERR CT
3672  015032  032737  000020  014242          BIT     #RNF,SRXES      ;SEEK ERROR?
3673  015040  001402                          BEQ     9$              ;BR IF NO
3674  015042  104067                          ERROR   67              ;SEEK ERROR READ SECTOR
3675  015044  000401                          BR      2$
3676  015046  104021                  9$:     ERROR   21              ;SOFT ERROR  READ SECTOR
3677
3678  015050  005037  002544          2$:     CLR     DOERR
3679  015054  023727  003260  000005          CMP     DXTST,#5        ;DOING WRITE DELETED DATA TESTS?
3680  015062  001005                          BNE     6$              ;BR IF NO
3681  015064  032737  000040  177172          BIT     #DD,RXES        ; 'DELETED DATA' SET?
3682  015072  001001                          BNE     6$              ;BR IF YES
3683  015074  104052                          ERROR   52              ;DD NOT SET
3684
3685  015076  000137  015102          6$:     JMP     DOEBUF          ;GO TO EMPTY BUFFER
3686
3687
3688
3689
3690
3691
3692                                  ;;*****************************************************************
3693                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX0
3694                                  ;& POINT TO THE CHECK DATA HANDLER.
3695                                  ;;*****************************************************************
3696
3697  015102  012703  000100          DOEBUF: MOV     #100,R3         ;WORD CT
3698  015106  012701  002564                  MOV     #DOBUF,R1       ;BUFFER ADDRESS
3699  015112  012737  015140  014230          MOV     #DOCDAT,DODISP  ;POINT TO CHK DATA AFT INTER.
3700  015120  012737  000103  177170          MOV     #<EBUF!IE>,RXCS ;ISSUE EMPTY BUFFER CMD
3701  015126  013721  177172          1$:     MOV     RXDB,(R1)+      ;GET WORD & STORE IT
3702  015132  005303                          DEC     R3
3703  015134  001374                          BNE     1$
3704  015136  000002                          RTI
3705
3706
```

```
3707                                    ;;*****************************************************************
3708                                    ; HANDLER TO CHECK DATA AFTER EMPTY BUFFER CMD ON DX0.
3709                                    ;WILL GO TO NEXT SECTOR HANDLER WHEN NO ERR OR HARD ERR.
3710                                    ;OR TO READ SECTOR (SAME) WHEN SOFT ERR.
3711                                    ;THIS ROUTINE IS ALSO ENTERED FROM A CRC ERROR FROM A READ COMMAND.
3712                                    ;;*****************************************************************
3713
3714  015140  005037  002552    DOCDAT: CLR     DOCMER
3715  015144  012703  000100            MOV     #100,R3         ;WORD CTR
3716  015150  013701  002546            MOV     DOTRK,R1
3717  015154  000301                    SWAB    R1
3718  015156  053701  002550            BIS     DOSEC,R1        ;R1 NOW HAS EXPECTED DATA
3719  015162  012702  002564            MOV     #DOBUF,R2       ;GET BUFFER ADDR
3720  015166  020112            1$:     CMP     R1,(R2)         ;COMPARE OK?
3721  015170  001407                    BEQ     2$              ;BR IF YES
3722  015172  010137  002556            MOV     R1,DOEXP        ;ELSE SAVE
3723  015176  011237  002560            MOV     (R2),DOREC
3724  015202  005237  002552            INC     DOCMER
3725  015206  000405                    BR      3$
3726
3727  015210  005303            2$:     DEC     R3
3728  015212  001403                    BEQ     3$
3729  015214  062702  000002            ADD     #2,R2
3730  015220  000762                    BR      1$
3731
3732  015222  005737  002552    3$:     TST     DOCMER          ;ANY COMP ERR?
3733  015226  001421                    BEQ     4$              ;BR IF NO
3734  015230  005737  002562            TST     DOCRC           ;HERE FROM CRC ERROR?
3735  015234  001402                    BEQ     14$             ;BR IF NO
3736  015236  104107                    ERROR   107             ;DATA CRC ERROR
3737  015240  000431                    BR      5$
3738
3739  015242  023727  002554  000012  14$:     CMP     DOCERR,#10.     ;ELSE, 10 ERRS YET?
3740  015250  001033                    BNE     7$              ;BR IF NO
3741  015252  023727  003260  000003            CMP     DXTST,#3        ;DOING INIT TEST?
3742  015260  001002                    BNE     8$              ;BR IN NO
3743  015262  104053                    ERROR   53              ;INITIALIZE ERROR
3744  015264  000417                    BR      5$
3745
3746  015266  104022            8$:     ERROR   22              ;HARD ERROR  DATA COMPARE
3747  015270  000415                    BR      5$
3748
3749  015272  005737  002562    4$:     TST     DOCRC           ;HERE FROM CRC ERROR?
3750  015276  001402                    BEQ     15$             ;BR IF NO
3751  015300  104110                    ERROR   110             ;CRC ERR WITH DATA OK
3752  015302  000410                    BR      5$
3753
3754  015304  005737  002554    15$:    TST     DOCERR          ;ANY PREV ERR?
3755  015310  001405                    BEQ     5$              ;BR IF NO
3756  015312  005237  027102            INC     TSERR           ;TOTAL SOFT ERR CT
3757  015316  005237  027106            INC     PSERR           ;PASS  SOFT ERR CT
3758  015322  104023                    ERROR   23              ;SOFT ERROR  DATA COMP
```

```
3759
3760  015324  005037  002554          5$:     CLR     DOCERR
3761  015330  005037  002562                  CLR     DOCRC
3762  015334  000137  015644                  JMP     DONEXT
3763
3764  015340  005237  002554          7$:     INC     DOCERR
3765  015344  023727  003260  000003          CMP     DXTST,#3        ;DOING INIT TEST?
3766  015352  001402                          BEQ     12$             ;BR IF YES
3767  015354  000137  014622                  JMP     DORSEC          ;ELSE READ SECTOR OVER AGAIN
3768
3769  015360  000137  016002          12$:    JMP     DODUN           ;ALL DONE
3770
3771
3772
3773                                  ;;**********************************************************************
3774                                  ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX0.
3775                                  ;WILL GO TO READ SECTOR (NEXT) IF ALL SECTORS ON TRACK NOT DONE.
3776                                  ;WILL GO TO FILL BUFFER (NEXT TRACK) IF ALL SECTORS ON TRACK DONE.
3777                                  ;WILL SHUT OFF DRV INTERRUPTS IF ALL TRACKS DONE, WITH A MESSAGE.
3778                                  ;;**********************************************************************
3779
3780  015364  005037  002544          DONXT': CLR     DOERR
3781  015370  005037  002552                  CLR     DOCMER
3782  015374  023727  002550  000031          CMP     DOSEC,#31       ;LAST ODD SECTOR?
3783  015402  001004                          BNE     1$              ;BR IF NO
3784  015404  012737  000002  002550          MOV     #2,DOSEC        ;ELSE DO EVEN SECTORS NOW
3785  015412  000407                          BR      2$
3786  015414  023727  002550  000032  1$:     CMP     DOSEC,#32       ;LAST EVEN SECTOR?
3787  015422  001405                          BEQ     3$              ;BR IF YES
3788  015424  062737  000002  002550          ADD     #2,DOSEC        ;ELSE BUMP SECTOR
3789  015432  000137  014622          2$:     JMP     DORSEC          ;GO READ SECTOR
3790
3791  015436  032777  010000  163474  3$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3792  015444  001411                          BEQ     6$
3793  015446  104401  020266                  TYPE    ,DRV0
3794  015452  104401  020721                  TYPE    ,TRK
3795  015456  013746  002546                  MOV     DOTRK,-(SP)     ;;SAVE DOTRK FOR TYPEOUT
3796  015462  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3797  015464  104401  021011                  TYPE    ,DUN
3798  015470  005237  003262          6$:     INC     FIN             ;SETUP TO GO TO DX1
3799  015474  023737  002546  003264          CMP     DOTRK,MAXTRK    ;LAST TRACK?
3800  015502  001406                          BEQ     4$              ;BR IF YES
3801  015504  005237  002546                  INC     DOTRK           ;BUMP TRACK
3802  015510  012737  000001  002550          MOV     #1,DOSEC        ;INIT SECTOR
3803  015516  000414                          BR      5$
3804
3805  015520  032777  010000  163412  4$:     BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3806  015526  001406                          BEQ     7$
3807  015530  104401  020266                  TYPE    ,DRV0
3808  015534  104401  020726                  TYPE    ,PATT
3809  015540  104401  021011                  TYPE    ,DUN
3810  015544  005237  002540          7$:     INC     DOPAT           ;PATT DONE FLAG
3811  015550  042737  000100  177170  5$:     BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
3812  015556  000002                          RTI
```

```
3813
3814                                    ;;*****************************************************************
3815                                    ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX0.
3816                                    ;;*****************************************************************
3817
3818  015560  005037  002544           DONXT2: CLR     DOERR
3819  015564  005037  002552                   CLR     DOCMER
3820  015570  005237  003262                   INC     FIN             ;SETUP TO GO TO DX1
3821  015574  005237  002542                   INC     DOCNT
3822  015600  023737  002542  003266           CMP     DOCNT,MAXSK     ;DID ALL SEEKS?
3823  015606  001012                           BNE     1$              ;BR IF NO
3824  015610  032777  010000  163322           BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
3825  015616  001406                           BEQ     1$
3826  015620  104401  020266                   TYPE    ,DRV0
3827  015624  104401  020741                   TYPE    ,RANDOM
3828  015630  104401  021011                   TYPE    ,DUN
3829  015634  042737  000100  177170  1$:      BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
3830  015642  000002                           RTI
3831
3832
3833
3834
3835
3836  015644  023727  003260  000001  DONEXT: CMP     DXTST,#1
3837  015652  001644                           BEQ     DONXT1
3838  015654  023727  003260  000002           CMP     DXTST,#2
3839  015662  001736                           BEQ     DONXT2
3840  015664  000446                           BR      DODUN
3841
```

```
3842
3843                                    ;;*****************************************************************
3844                                    ;HANDLER TO SETUP INITIALIZE COMMAND TEST
3845                                    ;;*****************************************************************
3846
3847  015666  012737  046032  177174  DOINIT: MOV     #46032,RXSA     ;TRY TO FAKE DRIVE TO GO TO TRK 76 SEC 26
3848                                            ;INIT SHOULD GO TO TRK 1, SEC 1.
3849  015674  012737  016002  014230           MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
3850  015702  012737  000111  177170           MOV     #<INITAL!IE>,RXCS       ;DO INIT COMMAND
3851  015710  000002                           RTI
3852
3853
3854                                    ;;*****************************************************************
3855                                    ;HANDLERS TO SETUP RESTORE COMMAND TEST
3856                                    ;;*****************************************************************
3857
3858  015712  012737  016002  014230  DOREST: MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
3859  015720  012737  000117  177170           MOV     #<RESTOR!IE>,RXCS
3860  015726  000002                           RTI
3861
3862  015730  012737  016016  014236  D1REST: MOV     #D1DUN,D1DISP
3863  015736  012737  000137  177170           MOV     #<RESTOR!IE!DX1>,RXCS
3864  015744  000002                           RTI
3865
3866                                    ;;*****************************************************************
3867                                    ;HANDLER TO SETUP READ STATUS COMMAND TEST
3868                                    ;;*****************************************************************
3869
3870  015746  012737  016002  014230  DOSTAT: MOV     #DODUN,DODISP   ;POINT TO DONE
3871  015754  012737  000113  177170           MOV     #<RSTAT!IE>,RXCS        ;DO READ STATUS COMMAND
3872  015762  000002                           RTI
3873
3874                                    ;;*****************************************************************
3875                                    ;HANDLER TO START INVALID ADDRESS TESTS
3876                                    ;;*****************************************************************
3877
3878  015764  012737  016002  014230  DOINV:  MOV     #DODUN,DODISP   ;POINT TO DONE
3879  015772  012737  000105  177170           MOV     #<WSEC!IE>,RXCS ;DO WRITE SECTOR COMMAND
3880  016000  000002                           RTI
3881
3882
3883                                    ;;*****************************************************************
3884                                    ;HANDLERS TO FINISH INITIALIZE, RESTORE, WRITE DELETED DATA,
3885                                    ;READ STATUS & INVALID ADDRESS TESTS.
3886                                    ;;*****************************************************************
3887
3888  016002  005237  003262           DODUN:  INC     FIN             ;SET DONE FLAG
3889  016006  042737  000100  177170           BIC     #IE,RXCS        ;DISABLE FURTHUR DX0 INTERRUPTS
3890  016014  000002                           RTI
3891
3892  016016  005237  003262           D1DUN:  INC     FIN
3893  016022  042737  000120  177170           BIC     #<IE DX1>,RXCS
3894  016030  000002                           RTI
3895
```

```
                                        ;;*****************************************************************
                                        ; HANDLER TO FILL THE DRIVE BUFFER FOR DX1
                                        ;& POINT TO THE WRITE SECTOR HANDLER.
                                        ;;*****************************************************************

016032  012703  000100          D1FBUF: MOV     #100,R3         ;WORD CT
016036  013701  003016                  MOV     D1TRK,R1
016042  000301                         SWAB    R1
016044  053701  003020                  BIS     D1SEC,R1

016050  005737  003272                  TST     COMP1           ;DONT SET BIT15 IN COMPATABILTIY TESTS
016054  001005                         BNE     2$
016056  005737  003274                  TST     COMP2
016062  001002                         BNE     2$
016064  052701  100000                  BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA

016070  012737  016116  014236  2$:     MOV     #D1WSEC,D1DISP  ;POINT TO WRITE SECTOR AFTER RTI
016076  012737  000121  177170          MOV     #<FBUF!IE!DX1>,RXCS     ;ISSUE FILL BUFFER CMD
016104  010137  177172          1$:     MOV     R1,RXDB         ;FILL ENTIRE SECTOR WITH ITS ADDRESS
016110  005303                         DEC     R3              ;& BIT 15 SET
016112  001374                         BNE     1$
016114  000002                         RTI

                                        ;;*****************************************************************
                                        ;HANDLER TO WRITE THE SECTOR FOR DX1 & POINT TO WRITE CHECK.
                                        ;;*****************************************************************

016116  013701  003016          D1WSEC: MOV     D1TRK,R1
016122  000301                         SWAB    R1
016124  053701  003020                  BIS     D1SEC,R1
016130  010137  177174                  MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
016134  012737  016152  014236          MOV     #D1WCHK,D1DISP  ;POINT TO WRITE CHECK HANDLER
016142  012737  000125  177170          MOV     #<WSEC!IE!DX1>,RXCS     ;ISSUE WRITE SECTOR COMMAND
016150  000002                         RTI
```

```
                                ;;*****************************************************************
                                ;HANDLER TO WRITE CHECK ON DX1.
                                ;WILL GO TO FILL BUFFER TO DO SAME SECTOR ON SOFT ERROR
                                ;OR NEXT SECTOR ON NO ERROR OR HARD ERROR.
                                ;WILL GO TO READ SECTOR AFTER ALL SECTORS ON TRACK ARE WRITTEN.
                                ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
                                ;;*****************************************************************

016152  005737  177170        D1WCHK: TST     RXCS            ;ERROR?
016156  100024                        BPL     1$              ;BR IN NO
016160  023727  003014  000012        CMP     D1ERR,#10.      ;10 ERRORS?
016166  001072                        BNE     6$              ;BR IF NO & TRY AGAIN
016170  004737  013064                JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
016174  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
016202  001410                        BEQ     7$              ;BR IF NO
016204  104070                        ERROR   70              ;SEEK ERROR WRITE SECTOR
016206  012737  016264  014236        MOV     #2$,D1DISP      ;RET TO 2$ AFT RESTORE
016214  012737  000137  177170        MOV     #<RESTOR!IE!DX1>,RXCS
016222  000002                        RTI

016224  104024                7$:     ERROR   24              ;ERROR WRITE SECTOR
016226  000416                        BR      2$              ;GO TO NEXT SECTOR

016230  005737  003014          1$:     TST     D1ERR           ;ANY PREV ERRORS?
016234  001413                        BEQ     2$              ;BR IF NO
016236  005237  027102                INC     TSERR           ;TOTAL SOFT ERR CT
016242  005237  027106                INC     PSERR           ;PASS  SOFT ERR CT
016246  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
016254  001402                        BEQ     8$              ;BR IF NO
016256  104071                        ERROR   71              ;SEEK ERROR WRITE SECTOR
016260  000401                        BR      2$
016262  104025                8$:     ERROR   25              ;ERROR WRITE SECTOR

016264  005037  003014          2$:     CLR     D1ERR
016270  023727  003020  000031        CMP     D1SEC,#31       ;LAST ODD SECTOR?
016276  001004                        BNE     3$              ;BR IF NO
016300  012737  000002  003020        MOV     #2,D1SEC        ;ELSE DO EVEN SECTORS NOW
016306  000407                        BR      4$
016310  023727  003020  000032  3$:     CMP     D1SEC,#32       ;LAST EVEN SECTOR?
016316  001405                        BEQ     5$              ;BR IF YES
016320  062737  000002  003020        ADD     #2,D1SEC        ;ELSE BUMP SECTOR
016326  000137  016032          4$:     JMP     D1FBUF          ;GO TO FILL BUFFER & DO ANOTHER

016332  012737  000001  003020  5$:     MOV     #1,D1SEC        ;ALL SECTORS DONE...CHECK DATA
016340  005037  003014                CLR     D1ERR
016344  005037  003022                CLR     D1CMER
016350  000137  016406                JMP     D1RSEC          ;GO TO READ SECTOR HANDLER

016354  005237  003014          6$:     INC     D1ERR
016360  032737  000020  014242        BIT     #RNF,SRXES      ;SEEK ERROR?
016366  001757                        BEQ     4$              ;BR IF NO
016370  012737  016032  014236        MOV     #D1FBUF,D1DISP  ;RET TO FILL BUFF AFT RESTORE
016376  012737  000137  177170        MOV     #<RESTOR!IE!DX1>,RXCS
016404  000002                        RTI
```

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

016406  013701  003016          D1RSEC: MOV     D1TRK,R1
016412  000301                          SWAB    R1
016414  053701  003020                  BIS     D1SEC,R1
016420  010137  177174                  MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR

016424  012737  016442  014236          MOV     #D1RCHK,D1DISP  ;POINT TO READ CHECK HANDLER
016432  012737  000127  177170          MOV     #<RSEC!IE!DX1>,RXCS     ;ISSUE READ SECTOR COMMAND
016440  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).
                                        ;;*****************************************************************

016442  005737  177170          D1RCHK: TST     RXCS            ;ERROR?
016446  100054                          BPL     1$              ;BR IF NO
016450  023727  003014  000012          CMP     D1ERR,#10.      ;10 ERRORS?
016456  001415                          BEQ     3$              ;BR IF YES
016460  005237  003014                  INC     D1ERR           ;ELSE TRY AGAIN
016464  032737  000020  014242          BIT     #RNF,SRXES      ;SEEK ERROR?
016472  001745                          BEQ     D1RSEC          ;BR IF NO
016474  012737  016406  014236          MOV     #D1RSEC,D1DISP  ;ELSE DO RESTORE & RET TO RD SEC
016502  012737  000137  177170          MOV     #<RESTOR!IE!DX1>,RXCS
016510  000002                          RTI

016512  004737  013064          3$:     JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
016516  032737  000020  014242          BIT     #RNF,SRXES      ;SEEK ERROR?
016524  001410                          BEQ     7$              ;BR IF NO
016526  104072                          ERROR   72              ;SEEK ERROR READ SECTOR

016530  012737  017376  014236          MOV     #D1NEXT,D1DISP  ;DO RESTORE
016536  012737  000137  177170          MOV     #<RESTOR!IE!DX1>,RXCS
016544  000002                          RTI

016546  032737  000010  014242  7$:     BIT     #CRC,SRXES      ;CRC ERROR?
016554  001404                          BEQ     10$             ;BR IF NO
016556  005237  003032                  INC     D1CRC           ;ELSE SET FLAG
016562  104111                          ERROR   111             ;HARD CRC ERROR
016564  000423                          BR      2$              ;& GO TO EMP BUFF TO CHK DATA

016566  005037  003032          10$:    CLR     D1CRC
016572  104026                          ERROR   26              ;HARD READ ERROR
016574  000137  017376                  JMP     D1NEXT
```

```
 4041
 4042  016600  005737  003014          1$:     TST     D1ERR           ;ANY PREV ERRORS?
 4043  016604  001413                          BEQ     2$              ;BR IF NO
 4044  016606  005237  027102                  INC     TSERR           ;TOTAL SOFT ERR CT
 4045  016612  005237  027106                  INC     PSERR           ;PASS  SOFT ERR CT
 4046  016616  032737  000020  014242          BIT     #RNF,SRXES      ;SEEK ERROR?
 4047  016624  001402                          BEQ     9$              ;BR IF NO
 4048  016626  104073                          ERROR   73              ;SEEK ERROR READ SECTOR
 4049  016630  000401                          BR      2$
 4050  016632  104027                  9$:     ERROR   27              ;SOFT ERROR  READ SECTOR
 4051
 4052  016634  005037  003014          2$:     CLR     D1ERR
 4053  016640  000137  016644                  JMP     D1EBUF          ;GO TO EMPTY BUFFER
 4054
 4055
 4056
 4057
 4058
 4059
 4060                                  ;;*****************************************************************
 4061                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX1
 4062                                  ;& POINT TO THE CHECK DATA HANDLER.
 4063                                  ;;*****************************************************************
 4064
 4065  016644  012703  000100          D1EBUF: MOV     #100,R3         ;WORD CT
 4066  016650  012701  003034                  MOV     #D1BUF,R1       ;BUFFER ADDRESS
 4067  016654  012737  016702  014236          MOV     #D1CDAT,D1DISP  ;POINT TO CHK DATA AFT INTER.
 4068  016662  012737  000123  177170          MOV     #<EBUF!IE!DX1>,RXCS     ;ISSUE EMPTY BUFFER CMD
 4069  016670  013721  177172          1$:     MOV     RXDB,(R1)+      ;GET WORD & STORE IT
 4070  016674  005303                          DEC     R3
 4071  016676  001374                          BNE     1$
 4072  016700  000002                          RTI
 4073
```

```
                                        ;;*****************************************************************
                                        ; HANDLER TO CHECK DATA AFTER EMPTY BUFFER CMD ON DX1.
                                        ;WILL GO TO NEXT SECTOR HANDLER WHEN NO ERR OR HARD ERR.
                                        ;OR TO READ SECTOR (SAME) WHEN SOFT ERR.
                                        ;THIS RCUTINE IS ALSO ENTERED FROM A CRC ERROR FROM A READ COMMAND.
                                        ;;*****************************************************************

016702  005037  003022        D1CDAT: CLR     D1CMER
016706  012703  000100                MOV     #100,R3         ;WORD CTR
016712  013701  003016                MOV     D1TRK,R1
016716  000301                        SWAB    R1
016720  053701  003020                BIS     D1SEC,R1

016724  005737  003272                TST     COMP1           ;DONT SET BIT15 IN COMPATABILITY TESTS
016730  001005                        BNE     6$
016732  005737  003274                TST     COMP2
016736  001002                        BNE     6$
016740  052701  100000                BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA

016744  012702  003034        6$:     MOV     #D1BUF,R2       ;GET BUFFER ADDR
016750  020112                1$:     CMP     R1,(R2)         ;COMPARE OK?
016752  001407                        BEQ     2$              ;BR IF YES
016754  010137  003026                MOV     R1,D1EXP        ;ELSE SAVE
016760  011237  003030                MOV     (R2),D1REC
016764  005237  003022                INC     D1CMER
016770  000405                        BR      3$
016772  005303                2$:     DEC     R3
016774  001403                        BEQ     3$
016776  062702  000002                ADD     #2,R2
017002  000762                        BR      1$

017004  005737  003022        3$:     TST     D1CMER          ;ANY COMP ERR?
017010  001413                        BEQ     4$              ;BR IF NO
017012  005737  003032                TST     D1CRC           ;HERE FROM CRC ERROR?
017016  001402                        BEQ     14$             ;BR IF NO
017020  104112                        ERROR   112             ;DATA CRC ERROR
017022  000423                        BR      5$
017024  023727  003024  000012  14$:  CMP     D1CERR,#10.     ;ELSE, 10 ERRS YET?
017032  001025                        BNE     7$              ;BR IF NO
017034  104030                        ERROR   30              ;HARD ERROR  DATA COMP
017036  000415                        BR      5$

017040  005737  003032        4$:     TST     D1CRC           ;HERE FROM CRC ERROR?
017044  001402                        BEQ     15$             ;BR IF NO
017046  104113                        ERROR   113             ;CRC ERR WITH DATA OK
017050  000410                        BR      5$
017052  005737  003024        15$:    TST     D1CERR          ;ANY ERR?
017056  001405                        BEQ     5$
017060  005237  027102                INC     TSERR           ;TOTAL SOFT ERR CT
017064  005237  027106                INC     PSERR           ;PASS  SOFT ERR CT
017070  104031                        ERROR   31              ;SOFT ERROR  DATA COMP
```

```
                                ;;*****************************************************************
                                ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX1.
                                ;WILL GO TO READ SECTOR (NEXT) IF ALL SECTORS ON TRACK NOT DONE.
                                ;WILL GO TO FILL BUFFER (NEXT TRACK) IF ALL SECTORS ON TRACK DONE.
                                ;WILL SHUT OFF DRV INTERRUPTS IF ALL TRACKS DONE, WITH A MESSAGE.
                                ;;*****************************************************************

017116  005037  003014        D1NXT1: CLR     D1ERR
017122  005037  003022                CLR     D1CMER
017126  023727  003020  000031        CMP     D1SEC,#31       ;LAST ODD SECTOR?
017134  001004                       BNE     1$              ;BR IF NO
017136  012737  000002  003020        MOV     #2,D1SEC        ;ELSE DO EVEN SECTORS NOW
017144  000407                       BR      2$
017146  023727  003020  000032  1$:   CMP     D1SEC,#32       ;LAST EVEN SECTOR?
017154  001405                       BEQ     3$              ;BR IF YES
017156  062737  000002  003020        ADD     #2,D1SEC        ;ELSE BUMP SECTOR
017164  000137  016406        2$:     JMP     D1RSEC          ;GO READ SECTOR

017170  032777  010000  161742  3$:   BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
017176  001411                       BEQ     6$
017200  104401  020273                TYPE    ,DRV1
017204  104401  020721                TYPE    ,TRK
017210  013746  003016                MOV     D1TRK,-(SP)     ;;SAVE D1TRK FOR TYPEOUT
017214  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
017216  104401  021011                TYPE    ,DUN
017222  005237  003262        6$:     INC     FIN             ;SETUP TO GO TO DX0
017226  023737  003016  003264        CMP     D1TRK,MAXTRK    ;LAST TRACK?
017234  001406                       BEQ     4$              ;BR IF YES
017236  005237  003016                INC     D1TRK           ;BUMP TRACK
017242  012737  000001  003020        MOV     #1,D1SEC        ;INIT SECTOR
017250  000414                       BR      5$

017252  032777  010000  161660  4$:   BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
017260  001406                       BEQ     7$
017262  104401  020273                TYPE    ,DRV1
017266  104401  020726                TYPE    ,PATT
017272  104401  021011                TYPE    ,DUN
017276  005237  003010        7$:     INC     D1PAT           ;PATT DONE FLAG
017302  042737  000120  177170  5$:   BIC     #<IE.DX1>,RXCS  ;DISABLE DX1 INTERRUPTS
017310  000002                       RTI
```

```
 4175
 4176                                   ;;*****************************************************************
 4177                                   ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX1.
 4178                                   ;;*****************************************************************
 4179
 4180  017312  005037  003014          D1NXT2: CLR     D1ERR
 4181  017316  005037  003022                  CLR     D1CMER
 4182  017322  005237  003262                  INC     FIN             ;SETUP TO GO TO DX0
 4183  017326  005237  003012                  INC     D1CNT
 4184  017332  023737  003012  003266          CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
 4185  017340  001012                          BNE     1$              ;BR IF NO
 4186  017342  032777  010000  161570          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
 4187  017350  001406                          BEQ     1$
 4188  017352  104401  020273                  TYPE    ,DRV1
 4189  017356  104401  020741                  TYPE    ,RANDOM
 4190  017362  104401  021011                  TYPE    ,DUN
 4191  017366  042737  000120  177170  1$:     BIC     #<IE!DX1>,RXCS  ;DISABLE DXi INTERRUPTS
 4192  017374  000002                          RTI
 4193
 4194
 4195
 4196
 4197  017376  023727  003260  000001  D1NEXT: CMP     DXTST,#1
 4198  017404  001644                          BEQ     D1NXT1
 4199  017406  000741                          BR      D1NXT2
 4200
```

```
4201
4202                                    ;;*****************************************************************
4203                                    ;THE FOLLOWING CODE, UP TO THE PROGRAM MESSAGES, IS A UTILITY TO COPY
4204                                    ;THE DISKETTE CONTAINING ONLY THE BOOT & THE EXERCISER FROM DX0 TO DX1.
4205                                    ;THE OPERATOR CAN TYPE 'P' TO DO ANOTHER COPY OR
4206                                    ;TYPE '240G' TO ENTER NORMAL TESTING.
4207                                    ;ENTIRE CODE, INCLUDING TEXT & ERROR TABLE, IS 400(10) WORDS.
4208                                    ;;*****************************************************************
4209
4210  017410  005037  003270        COPY:   CLR     COPFLG                  ;DONT NEED FLAG ANY MORE
4211  017414  104401  021210                TYPE    ,COPMSG                 ;INSERT SCRATCH IN DX1 & TYPE P
4212  017420  000000                        HALT                            ;WAIT FOR PROCEED
4213
4214  017422  004737  017730                JSR     PC,GETSEC               ;CALCULATE MAX SECTORS TO BE COPIED
4215
4216  017426  005037  017722                CLR     ERFLG                   ;INITIALIZE
4217  017432  012737  000001  017724        MOV     #1,TRACK
4218  017440  012737  000001  017726        MOV     #1,SECT
4219  017446  105737  177170                TSTB    RXCS                    ;DONE?  (CTL RDY)
4220  017452  100375                        BPL     .-4                     ;BR IF NO & TRY AGAIN
4221
4222  017454  004537  020000        1$:     JSR     R5,RDWR         ;READ DX0
4223  017460  000007                           RSEC
4224
4225  017462  005737  177170                TST     RXCS                    ;ERROR?
4226  017466  100011                        BPL     2$                      ;BR IF NO
4227  017470  005237  017722                INC     ERFLG
4228  017474  023727  017722  000012        CMP     ERFLG,#10.              ;10 ERRORS?
4229  017502  001364                        BNE     1$                      ;BR IF NO & TRY AGAIN
4230  017504  104100                        ERROR   100                     ;HARD ERR READ SECTOR DX0
4231  017506  000000                        HALT
4232  017510  000776                        BR      .-2                     ;FORCE A RESTART FROM 250
4233
4234  017512  005037  017722        2$:     CLR     ERFLG
4235  017516  004537  020040                JSR     R5,EBUFF        ;EMPTY BUFFER INTO DOBUF
4236  017522  002564                           DOBUF
4237
4238  017524  004537  020000        3$:     JSR     R5,RDWR         ;WRITE DX1
4239  017530  000025                           <WSEC!DX1>
4240
4241  017532  005737  177170                TST     RXCS                    ;ERROR?
4242  017536  100011                        BPL     4$                      ;BR IF NO
4243  017540  005237  017722                INC     ERFLG
4244  017544  023727  017722  000012        CMP     ERFLG,#10.              ;10 ERRORS?
4245  017552  001364                        BNE     3$                      ;BR IF NO & TRY AGAIN
4246  017554  104101                        ERROR   101                     ;ERROR WRITE SECTOR
4247  017556  000000                        HALT
4248  017560  000776                        BR      .-2                     ;FORCE RESTART AT 250
```

```
4249
4250  017562  005037  017722          4$:     CLR     ERFLG
4251  017566  004537  020600                  JSR     R5,RDWR         ;READ DX1
4252  017572  000027                             <RSEC!DX1>
4253
4254  017574  005737  177170                  TST     RXCS                    ;ERROR?
4255  017600  100011                          BPL     5$                      ;BR IF NO
4256  017602  005237  017722                  INC     ERFLG
4257  017606  023727  017722  000012          CMP     ERFLG,#10.              ;10 ERRORS?
4258  017614  001362                          BNE     4$                      ;BR IF NO & TRY AGAIN
4259  017616  104103                          ERROR   103                     ;HARD ERR READ SECTOR DX1
4260  017620  000000                          HALT
4261  017622  000776                          BR      .-2                     ;FORCE 250 RESTART
4262
4263  017624  004537  020040          5$:     JSR     R5,EBUFF        ;EMPTY BUFFER INTO D1BUF
4264  017630  003034                             D1BUF
4265
4266  017632  004737  020102                  JSR     PC,DATCHK       ;COMPARE DOBUF WITH D1BUF
4267
4268  017636  005237  017720                  INC     SECCNT                  ;TOTAL DONE
4269
4270  017642  023737  017720  017716          CMP     SECCNT,MAXSEC           ;DID TOTAL # SECTORS?
4271  017650  001415                          BEQ     8$                      ;BR IF YES
4272
4273  017652  023727  017726  000032          CMP     SECT,#32                ;ELSE DID WE DO LAST SECTOR ON TRACK?
4274  017660  001403                          BEQ     6$                      ;BR IF YES
4275  017662  005237  017726                  INC     SECT                    ;ELSE BUMP & DO ANOTHER
4276  017666  000672                          BR      1$
4277
4278  017670  005237  017724          6$:     INC     TRACK                   ;BUMP TRK
4279  017674  012737  000001  017726          MOV     #1,SECT                 ;INIT SECTOR
4280  017702  000664                          BR      1$                      ;& DO ANOTHER
4281
4282  017704  104401  021311          8$:     TYPE    ,COPDUN                 ;DONE MSG
4283  017710  000000                          HALT
4284  017712  000137  017410                  JMP     COPY                    ;DO AGAIN IF 'P' TYPED
4285
4286  017716  000000                  MAXSEC: 0                               ;TOTAL # SECTORS TO READ FROM DX0
4287  017720  000000                  SECCNT: 0                               ;TOTAL # SECTORS WRITTEN TO DX1
4288  017722  000000                  ERFLG:  0
4289  017724  000000                  TRACK:  0
4290  017726  000000                  SECT:   0
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO CALCULATE THE MAX # OF SECTORS TO READ FROM DX0
                                ;;*****************************************************************

017730  012700  027744  GETSEC: MOV     #LSTAD,R0               ;GET MAX ADDR
017734  006200                  ASR     R0                      ;GET WORD CT
017736  005500                  ADC     R0                      ;DONT LOOSE ANY
017740  062700  000377          ADD     #377,R0
017744  042700  000377          BIC     #377,R0                 ;ROUND OFF TO NEAREST WHOLE WORD

017750  000300                  SWAB    R0
017752  006300                  ASL     R0
017754  006300                  ASL     R0                      ;DIVIDE BY 100(8) FOR SECTOR CT.

017756  062700  000074          ADD     #<15.*4>,R0             ;ADD SECTOR CT FOR BLOCKS 0 - 14 OF DX0
017762  062700  000010          ADD     #8.,R0                  ;2 MORE BLOCKS (NECESSARY FUDGE FACTOR)

017766  010037  017716          MOV     R0,MAXSEC               ;SAVE
                                                                ;THIS IS THE MAX SECTORS WE WILL READ OFF DX0
                                                                ;& WRITE TO DX1 STARTING AT TRK 1, SECT 1
017772  005037  017720          CLR     SECCNT                  ;COUNT OF TOTAL SECTORS WRITTEN TO DX1
017776  000207                  RTS     PC


                                ;;*****************************************************************
                                ;ROUTINE TO READ OR WRITE DX0 OR DX1
                                ;R5 POINTS TO THE COMMAND & DISK #
                                ;;*****************************************************************

020000  013700  017724  RDWR:   MOV     TRACK,R0
020004  000300                  SWAB    R0
020006  053700  017726          BIS     SECT,R0
020012  010037  177174          MOV     R0,RXSA                 ;LOAD TRK/SEC
020016  012537  177170          MOV     (R5)+,RXCS              ;ISSUE COMMAND
020022  004537  012634          JSR     R5,TIMER2               ;WAIT FOR DONE
020026  177170                      RXCS
020030  000200                      DONE
020032  104124                  ERROR   124                     ;NO DONE
020034  000000                  HALT
020036  000205                  RTS     R5
```

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

020040  012700  000100        EBUFF:  MOV     #100,R0                 ;WD CT
020044  012501                        MOV     (R5)+,R1                ;BUFFER ADDR
020046  012737  000003  177170        MOV     #EBUF,RXCS              ;ISSUE EMPTY BUFFER COMMAND
020054  013721  177172        1$:     MOV     RXDB,(R1)+              ;PUT IT IN TABLE
020060  005300                        DEC     R0
020062  001374                        BNE     1$
020064  004537  012634                JSR     R5,TIMER2               ;WAIT FOR DONE
020070  177170                           RXCS
020072  000200                           DONE
020074  104125                        ERROR   125                     ;NO DONE
020076  000000                        HALT
020100  000205                        RTS     R5


                                ;;*****************************************************************
                                ;ROUTINE TO COMPARE DOBUF WITH D1BUF
                                ;;*****************************************************************

020102  012700  000100        DATCHK: MOV     #100,R0                 ;WD CT
020106  012701  002564                MOV     #DOBUF,R1
020112  012702  003034                MOV     #D1BUF,P2
020116  022122                1$:     CMP     (R1)+,(R2)+             ;COMPARE?
020120  001403                        BEQ     2$                      ;BR IF YES
020122  104102                        ERROR   102                     ;MISCOMPARE
020124  000000                        HALT
020126  000776                        BR      .-2

020130  005300                2$:     DEC     R0
020132  001371                        BNE     1$
020134  000207                        RTS     PC
```

```
                                .SBTTL  PROGRAM MESSAGES

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

                                .NLIST  BEX

020136  041600  052514  052123  CLOPT:  .ASCIZ  <CRLF>/CLUSTER OPTION/
020156  052200  051105  020115  CLT1:   .ASCIZ  <CRLF>/TERM #1/
020167     200  042524  046522  CLT2:   .ASCIZ  <CRLF>/TERM #2/
020200  052200  051105  020115  CLT3:   .ASCIZ  <CRLF>/TERM #3/
020211     200  054523  041516  SCOM:   .ASCIZ  <CRLF>/SYNC COMM/
020224  040600  054523  041516  ACOM:   .ASCIZ  <CRLF>/ASYNC COMM/
020240  050200  044522  052116  PRINT:  .ASCIZ  <CRLF>/PRINTER/
020251     040  047520  052122  PORT:   .ASCIZ  / PORT TESTED/
020266  042200  030130     000  DRV0:   .ASCIZ  <CRLF>/DX0/
020273     200  054104  000061  DRV1:   .ASCIZ  <CRLF>/DX1/
020300  041600  047514  045503  CLOCK:  .ASCIZ  <CRLF>/CLOCK/
020307     200  044505  026523  EISTXT: .ASCIZ  <CRLF>/EIS-FIS OPTION/
020327     113  046440  046505  MEM:    .ASCIZ  /K MEMORY PRESENT/


020350  047040  052117  050040  NOTPRES:        .ASCIZ  / NOT PRESENT/
020365     200  044412  051516  WARNING:        .ASCII  <CRLF><LF>/INSERT SCRATCH DISKS, TYPE 'P' FOR NORMAL TESTING/
020450  023600  032062  043460          .ASCII  <CRLF>/'240G' FOR NORMAL RESTARTS/
020503     200  031047  030065          .ASCII  <CRLF>/'250G' TO COPY SYS EXERCISER DISK/
020545     200  031047  030066          .ASCII  <CRLF>/'260G' FOR COMPATABILITY PASS 1: WRITE/
020614  023600  033462  043460          .ASCIZ  <CRLF>/'270G' FOR COMPATABILITY PASS 2: READ/
020663     040  042524  052123  DROP:   .ASCIZ  / TESTING DROPPED/
020704  051040  047125  044516  RUN:    .ASCIZ  / RUNNING/
020715     040  045517     000  OK:     .ASCIZ  / OK/
020721     040  051124  000113  TRK:    .ASCIZ  / TRK/
020726  042040  052101  020101  PATT:   .ASCIZ  / DATA PATT/
020741     040  040522  042116  RANDOM: .ASCIZ  / RANDOM SEEKS/
020757     200  044505  026523  FISEIS: .ASCIZ  <CRLF>/EIS-FIS TESTS/
020776  046600  046505  052040  MEMORY: .ASCIZ  <CRLF>/MEM TESTS/
021011     040  047504  042516  DUN:    .ASCIZ  / DONE/
021017     200  042412  042116  ENDPAS: .ASCIZ  <CRLF><LF>/END PASS #/
021034  052011  052117  046101  MSG1:   .ASCIZ  /       TOTAL ERRORS:          /
021063     200  004411  052011  MSG2:   .ASCIZ  <CRLF>/                TOTAL SOFT ERRORS:     /
021117     200  004412  004411  MSG3:   .ASCIZ  <CRLF><LF>/                  TOTAL ERRORS THIS PASS:/
021154  004600  004411  047523  MSG4:   .ASCIZ  <CRLF>/                SOFT  ERRORS THIS PASS:/
021210  041600  050117  020131  COPMSG: .ASCII  <CRLF>/COPY DX0 TO DX1/
021230  044600  051516  051105          .ASCIZ  <CRLF>/INSERT SCRATCH DISK IN DX1, TYPE 'P' TO PROCEED/
021311     200  047504  042516  COPDUN: .ASCIZ  <CRLF>/DONE...TYPE 'P' TO DO AGAIN OR '240G' FOR NORMAL TESTING/
021403     200  041412  046517  COMPAT: .ASCII  <CRLF><LF>/COMPATABILITY PASS 1 (WRITE) DONE/
021446  042200  020117  050047          .ASCIZ  <CRLF>/DO 'P' FOR PASS 2 (READ)/
021500  042200  053105  020115  DEVM:   .ASCIZ  <CRLF>/DEVM = /
021511     115  046505  051117  EM1:    .ASCIZ  /MEMORY TIMEOUT/
021530  042515  020115  040504  EM2:    .ASCIZ  /MEM DATA COMP ERR/
021552  054523  041516  041440  EM3:    .ASCIZ  /SYNC COMM DID NOT RECEIVE SYNC CHAR/
021616  051120  047111  042524  EM4:    .ASCIZ  /PRINTER STATUS ERR/
021641     124  051105  020115  EM5:    .ASCIZ  /TERM #1 DATA COMP ERR/
021667     124  051105  020115  EM6:    .ASCIZ  /TERM #2 DATA COMP ERR/
```

```
021715     124  051105  020115  EM7:    .ASCIZ  /TERM #3 DATA COMP ERR/
021743     101  054523  041516  EM8:    .ASCIZ  /ASYNC COMM DATA COMP ERR/
021774  054523  041516  041440  EM9:    .ASCIZ  /SYNC COMM DATA COMP ERR/
022024  047125  054105  020120  EM10:   .ASCIZ  /UNEXP DISK INTERRUPT/
022051     104  030130  053440  EM11:   .ASCIZ  /DX0 WRITE ERR/
022067     104  030130  044040  EM13:   .ASCIZ  /DX0 HARD ERR - READ SECT/
022120  054104  020060  047523  EM14:   .ASCIZ  /DX0 SOFT ERR - READ SECT/
022151     104  030130  044040  EM15:   .ASCIZ  /DX0 HARD ERR - DATA COMP/
022202  054104  020060  047523  EM16:   .ASCIZ  /DX0 SOFT ERR - DATA COMP/
022233     104  030530  053440  EM17:   .ASCIZ  /DX1 WRITE ERR/
022251     104  030530  044040  EM19:   .ASCIZ  /DX1 HARD ERR - READ SECT/
022302  054104  020061  047523  EM20:   .ASCIZ  /DX1 SOFT ERR - READ SECT/
022333     104  030530  044040  EM21:   .ASCIZ  /DX1 HARD ERR - DATA COMP/
022364  054104  020061  047523  EM22:   .ASCIZ  /DX1 SOFT ERR - DATA COMP/
022415     103  045514  044040  EM23:   .ASCIZ  /CLK HUNG/
022426  051120  047111  042524  EM24:   .ASCIZ  /PRINTER HUNG/
022443     124  051105  020115  EM25:   .ASCIZ  /TERM #1 HUNG/
022460  042524  046522  021440  EM26:   .ASCIZ  /TERM #2 HUNG/
022475     124  051105  020115  EM27:   .ASCIZ  /TERM #3 HUNG/
022512  054523  041516  041440  EM28:   .ASCIZ  /SYNC COMM HUNG/
022531     101  054523  041516  EM29:   .ASCIZ  /ASYNC COMM HUNG/
022551     104  030130  044040  EM30:   .ASCIZ  /DX0 HUNG/
022562  054104  020061  052510  EM31:   .ASCIZ  /DX1 HUNG/
022573     123  047131  020103  EM32:   .ASCIZ  /SYNC COMM - DATA NOT AVAIL NOT SET/
022636  054523  041516  041440  EM33:   .ASCIZ  /SYNC COMM - RECVR ACTIVE NOT SET/
022677     123  047131  020103  EM34:   .ASCIZ  /SYNC COMM - RECVR DONE NOT SET/
022736  054104  020060  042047  EM35:   .ASCIZ  /DX0 'DEL DATA' BIT 5 NOT SET IN RXES/
023003     104  030130  044440  EM36:   .ASCIZ  /DX0 INIT CMD DID NOT READ TRK 1, SEC 1/
023052  054104  020060  042522  EM37:   .ASCIZ  /DX0 RESTORE CMD ERR/
023076  054104  020060  044447  EM38:   .ASCIZ  /DX0 'INIT DONE' BIT 0 NOT SET IN RXES/
023144  054104  020060  042522  EM39:   .ASCIZ  /DX0 READ STATUS CMD ERR/
023174  054104  020060  047111  EM40:   .ASCIZ  /DX0 INV ADDR BIT 1 NOT SET IN RXES/
023237     104  030130  042440  EM41:   .ASCIZ  /DX0 ERR BIT NOT SET IN RXCS/
023273     104  030130  044440  EM42:   .ASCIZ  /DX0 INIT CMD ERR/
023314  054104  020060  042523  EM43:   .ASCIZ  /DX0 SEEK ERR - WRITE SECT/
023346  054104  020060  042523  EM45:   .ASCIZ  /DX0 SEEK ERR - READ SECT/
023377     104  030530  051440  EM47:   .ASCIZ  /DX1 SEEK ERR - WRITE SECT/
023431     104  030530  051440  EM49:   .ASCIZ  /DX1 SEEK ERR - READ SECT/
023462  054104  020061  042522  EM51:   .ASCIZ  /DX1 RESTORE CMD ERR/
023506  044504  045523  023440  EM52:   .ASCIZ  /DISK 'DONE' NOT SET/
023532  044506  046114  023040  EM53:   .ASCIZ  /FILL & EMPTY BUFF TEST... DATA MISCOMP/
023601     104  052101  020101  EM54:   .ASCIZ  /DATA MISCOMP/
023616  042522  046120  041501  EM55:   .ASCIZ  /REPLACE DX0...10 BAD SECTORS/
023653     122  050105  040514  EM56:   .ASCIZ  /REPLACE DX1...10 BAD SECTORS/
023710  054104  020060  040510  EM57:   .ASCIZ  /DX0 HARD CRC ERR/
023731     104  030130  042040  EM58:   .ASCIZ  /DX0 DATA CRC ERR/
023752  054104  020060  051120  EM59:   .ASCIZ  /DX0 PREV CRC ERR WITH DATA OK/
024010  054104  020061  040510  EM60:   .ASCIZ  /DX1 HARD CRC ERR/
024031     104  030530  042040  EM61:   .ASCIZ  /DX1 DATA CRC ERR/
024052  054104  020061  051120  EM62:   .ASCIZ  /DX1 PREV CRC ERR WITH DATA OK/
024110  044505  026523  044506  EM63:   .ASCIZ  /EIS-FIS CONDITION CODE ERROR/
024145     105  051511  043055  EM64:   .ASCIZ  /EIS-FIS DATA ERROR/

024170  051105  047522  020122  DH1:    .ASCIZ  /ERROR #        ERR PC/
024207     105  051122  051117  DH2:    .ASCIZ  /ERROR #        ERR PC  ADDR/
024233     105  051122  051117  DH3:    .ASCIZ  /ERROR #        ERR PC  ADDR    EXPECT  RECVD/
```

```
024275     105  051122  051117  DH4:    .ASCIZ  /ERROR #        ERR PC  PR STATUS/
024326  051105  047522  020122  DH5:    .ASCIZ  /ERROR #        ERR PC  EXPECT  RECVD/
024363     105  051122  051117  DH6:    .ASCIZ  /ERROR #        DXTST#  ERR PC  RXCS    RXES    RXSA/
024430  051105  047522  020122  DH7:    .ASCIZ  /ERROR #        DXTST#  ERR PC  RXCS    RXES    TRACK   SECTOR  # RETRIE
024517     104  052130  052123  DH8:    .ASCIZ  /DXTST #        ERR PC  RXCS    RXES    RXSA    EXPECT  RECVD   # RETRIES

                                 .EVEN

024606  027076  001116  000000  DT1:    .WORD   ERRNUM,$ERRPC,0
024614  027076  001116  004546  DT2:    .WORD   ERRNUM,$ERRPC,ADDR,0
024624  027076  001116  004546  DT3:    .WORD   ERRNUM,$ERRPC,ADDR,PAT,MEMHLD,0
024640  027076  001116  013336  DT4:    .WORD   ERRNUM,$ERRPC,SPRS,0
024650  027076  001116  013434  DT5:    .WORD   ERRNUM,$ERRPC,T1CHR,T1HLD,0
024662  027076  001116  013534  DT6:    .WORD   ERRNUM,$ERRPC,T2CHR,T2HLD,0
024674  027076  001116  013634  DT7:    .WORD   ERRNUM,$ERRPC,T3CHR,T3HLD,0
024706  027076  001116  013734  DT8:    .WORD   ERRNUM,$ERRPC,COMCHR,COMHLD,0
024720  027076  003260  001116  DT9:    .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,SRXSA,0
024736  027076  003260  001116  DT10:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D0TRK,D0SEC,D0ERR,0
024760  003260  001116  177170  DT11:   .WORD   DXTST,$ERRPC,RXCS,RXES,RXSA,D0EXP,D0REC,D0CERR,0
025002  027076  003260  001116  DT12:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D1TRK,D1SEC,D1ERR,0
025024  003260  001116  177170  DT13:   .WORD   DXTST,$ERRPC,RXCS,RXES,RXSA,D1EXP,D1REC,D1CERR,0
025046  027076  003260  001116  DT14:   .WORD   ERRNUM,DXTST,$ERRPC,RXCS,RXES,RXSA,0
025064  027076  001116  007242  DT15:   .WORD   ERRNUM,$ERRPC,EXP0,EBHLD,0
025076  027076  001116  007244  DT16:   .WORD   ERRNUM,$ERRPC,EXP1,EBHLD,0

025110  000000                  DF:     .WORD   0               ;OCTAL FORMAT FOR ALL
025112  000000                          .WORD   0
025114  000000                          .WORD   0

025116  000000                  DF1:    .WORD   0               ;OCTAL,OCTAL
025120  000000                          .WORD   0               ;OCTAL,OCTAL
025122  000400                          .WORD   400             ;DECIMAL,OCTAL
025124  000401                          .WORD   401             ;DECIMAL,DECIMAL

025126  000000                  DF2:    .WORD   0               ;OCTAL,OCTAL
025130  000000                          .WORD   0               ;OCTAL,OCTAL
025132  000000                          .WORD   0               ;OCTAL,OCTAL
025134  000400                          .WORD   400             ;DECIMAL,OCTAL

                                 .LIST   BEX
```

```
                                .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
                                ;*

025136  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
025142  100002                          BPL     1$              ;;BR IF YES
025144  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
025146  000430                          BR      3$              ;;LEAVE
025150  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
025152  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
025156  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
025164  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
025166  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
025174  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
025176  010037  025206                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
025202  004737  026270                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
025206  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
025210  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
025216  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
025220  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
025222  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
025224  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
025226  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
025230  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
025234  000002                          RTI                     ;;RETURN
025236  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
025242  001430                          BEQ     8$
025244  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
025250  001006                          BNE     5$
025252  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
025254  104401                          TYPE                    ;;TYPE A CR AND LF
025256  001165                          $CRLF
025260  105037  025414                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
025264  000755                          BR      2$              ;;GET NEXT CHARACTER
025266  004737  025350          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
025272  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
025276  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
025300  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
025304  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
025310  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
025312  004737  025350                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
025316  105337  025414                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
 4434  025322  000770                        BR      7$              ;;LOOP
 4435
 4436                                  ;HORIZONTAL TAB PROCESSOR
 4437
 4438  025324  112716  000040          8$:   MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 4439  025330  004737  025350          9$:   JSR     PC,$TYPEC       ;;TYPE A SPACE
 4440  025334  132737  000007  025414        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 4441  025342  001372                        BNE     9$              ;;TAB STOP
 4442  025344  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
 4443  025346  000724                        BR      2$              ;;GET NEXT CHARACTER
 4444  025350  105777  153574          $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 4445  025354  100375                        BPL     $TYPEC
 4446  025356  116677  000002  153566        MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 4447  025364  122766  000015  000002        CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 4448  025372  001003                        BNE     1$              ;;BRANCH IF NO
 4449  025374  105037  025414                CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 4450  025400  000406                        BR      $TYPEX          ;;EXIT
 4451  025402  122766  000012  000002  1$:   CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 4452  025410  001402                        BEQ     $TYPEX          ;;BRANCH IF YES
 4453  025412  105227                        INCB    (PC)+           ;;COUNT THE CHARACTER
 4454  025414  000000          $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 4455  025416  000207          $TYPEX: RTS     PC
 4456
```

```
                                        .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.
025420  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
025426  001114                        BNE     15$             ;;BRANCH IF NO
025430  105777  153510                TSTB    @$TKS           ;;CHAR THERE?
025434  100111                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
025436  117746  153504                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
025442  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
025446  022726  000007                CMP     #7,(SP)+        ;;IS IT A CON   L G?
025452  001102                        BNE     15$             ;;NO, RETURN TO USER
025454  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
025462  001476                        BEQ     15$             ;;BRANCH IF YES

025464  104401  026232                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
025470  104401  026237        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
025474  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
025500  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
025502  104401  026250                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
025506  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
025510  005046                        CLR     -(SP)           ;;THE NEW SWR
025512  105777  153426          7$:   TSTB    @$TKS           ;;CHAR THERE?
025516  100375                        BPL     7$              ;;IF NOT TRY AGAIN

025520  117746  153422                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
025524  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

025530  021627  000003                CMP     (SP),#3         ;;IS IT A CONTROL-C?
025534  001015                        BNE     9$              ;;BRANCH IF NOT
025536  104401  026220                TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
025542  062706  000006                ADD     #6,SP           ;;CLEAN UP STACK
025546  123727  001135  000001        CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
025554  001003                        BNE     8$              ;;BRANCH IF NO
025556  012777  000100  153360        MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
025564  000137  012152          8$:   JMP     GT$DEV          ;;CONTROL-C RESTART

025570  021627  000025          9$:   CMP     (SP),#25        ;;IS IT A CONTROL-U?
025574  001005                        BNE     10$             ;;BRANCH IF NOT
025576  104401  026225                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
025602  062706  000006          20$:  ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
025606  000737                        BR      19$             ;;LET'S TRY IT AGAIN

025610  021627  000015          10$:  CMP     (SP),#15        ;;IS IT A <CR>?
025614  001022                        BNE     16$             ;;BRANCH IF NO
025616  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
025622  001403                        BEQ     11$             ;;BRANCH IF YES
025624  016677  000002  153306        MOV     2(SP),@SWR      ;;SAVE NEW SWR
```

```
 4513  025632  062706  000006            11$:    ADD     #6,SP           ;;CLEAR UP STACK
 4514  025636  104401  001165            14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 4515  025642  123727  001135  000001            CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 4516  025650  001003                            BNE     15$             ;;BRANCH IF NOT
 4517  025652  012777  000100  153264            MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 4518  025660  000002                    15$:    RTI                     ;;RETURN
 4519  025662  004737  025350            16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 4520  025666  021627  000060                    CMP     (SP),#60        ;;CHAR < 0?
 4521  025672  002420                            BLT     18$             ;;BRANCH IF YES
 4522  025674  021627  000067                    CMP     (SP),#67        ;;CHAR > 7?
 4523  025700  003015                            BGT     18$             ;;BRANCH IF YES
 4524  025702  042726  000060                    BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 4525  025706  005766  000002                    TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4526  025712  001403                            BEQ     17$             ;;BRANCH IF YES
 4527  025714  006316                            ASL     (SP)            ;;NO, SHIFT PRESENT
 4528  025716  006316                            ASL     (SP)            ;;   CHAR OVER TO MAKE
 4529  025720  006316                            ASL     (SP)            ;;   ROOM FOR NEW ONE.
 4530  025722  005266  000002            17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 4531  025726  056616  177776                    BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4532  025732  000667                            BR      7$              ;;GET THE NEXT ONE
 4533  025734  104401  001164            18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 4534  025740  000720                            BR      20$             ;;SIMULATE CONTROL-U
 4535                                    .DSABL  LSB
 4536
 4537
 4538                                    ;;*****************************************************************
 4539                                    ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4540                                    ;*CALL:
 4541                                    ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 4542                                    ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4543                                    ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4544                                    ;
 4545
 4546  025742  011646                    $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 4547  025744  016666  000004  000002            MOV     4(SP),2(SP)     ;;SAVE THE PS
 4548  025752  105777  153166            1$:     TSTB    @$TKS           ;;WAIT FOR
 4549  025756  100375                            BPL     1$              ;;A CHARACTER
 4550  025760  117766  153162  000004            MOVB    @$TKB,4(SP)     ;;READ THE TTY
 4551  025766  042766  177600  000004            BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4552  025774  026627  000004  000023            CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4553  026002  001013                            BNE     3$              ;;BRANCH IF NO
 4554  026004  105777  153134            2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 4555  026010  100375                            BPL     2$              ;;LOOP UNTIL ITS THERE
 4556  026012  117746  153130                    MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 4557  026016  042716  177600                    BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4558  026022  022627  000021                    CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4559  026026  001366                            BNE     2$              ;;IF NOT DISCARD IT
 4560  026030  000750                            BR      1$              ;;YES, RESUME
 4561  026032  026627  000004  000140    3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4562  026040  002407                            BLT     4$              ;;BRANCH IF YES
 4563  026042  026627  000004  000175            CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4564  026050  003003                            BGT     4$              ;;BRANCH IF YES
 4565  026052  042766  000040  000004            BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 4566  026060  000002                    4$:     RTI                     ;;GO BACK TO USER
 4567                                    ;;*****************************************************************
 4568                                    ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
```

```
                                        ;*CALL:
                                        ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
                                        ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
                                        ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S

026062  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
026064  012703  026210  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
026070  022703  026220  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
026074  101415                  BLOS    4$              ;;BR IF YES
026076  104410                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
026100  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
026102  122713  000003          CMPB    #3,(R3)         ;;IS IT A CONTROL-C?
026106  001005                  BNE     10$             ;;BRANCH IF NO
026110  104401  026220          TYPE    ,$CNTLC         ;;TYPE A CONTROL-C (^C)
026114  012603                  MOV     (SP)+,R3        ;;RESTORE R3
026116  000137  012152          JMP     GT$DEV          ;;GOTO CONTROL-C RESTART
026122  122713  000177  10$:     CMPB    #177,(R3)       ;;IS IT A RUBOUT
026126  001003                  BNE     3$              ;;SKIP IF NOT
026130  104401  001164  4$:     TYPE    ,$QUES          ;;TYPE A '?'
026134  000753                  BR      1$              ;;CLEAR THE BUFFER AND LOOP
026136  111337  026206  3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
026142  104401  026206          TYPE    ,9$
026146  122723  000015          CMPB    #15,(R3)+       ;;CHECK FOR RETURN
026152  001346                  BNE     2$              ;;LOOP IF NOT RETURN
026154  105063  177777          CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
026160  104401  001166          TYPE    ,$LF            ;;TYPE A LINE FEED
026164  012603                  MOV     (SP)+,R3        ;;RESTORE R3
026166  011646                  MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
026170  016666  000004  000002  MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
026176  012766  026210  000004  MOV     #$TTYIN,4(SP)
026204  000002                  RTI                     ;;RETURN
026206     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
026207     000                  .BYTE   0               ;;TERMINATOR
026210  000010          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
026220  041536  005015     000  $CNTLC: .ASCIZ  /^C/<15><12>    ;;CONTROL "C"
026225     136  006525  000012  $CNTLU: .ASCIZ  /^U/<15><12>    ;;CONTROL "U"
026232  043536  005015     000  $CNTLG: .ASCIZ  /^G/<15><12>    ;;CONTROL "G"
026237     015  051412  051127  $MSWR:  .ASCIZ  <15><12>/SWR = /
026244  036440  000040
026250  020040  042516  020127  $MNEW:  .ASCIZ  /  NEW = /
026256  020075     000
        026262                  .EVEN
```

```
                        .SBTTL  APT COMMUNICATIONS ROUTINE
                        ;;*****************************************************************
026262  112737  000001  026526  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
026270  112737  000001  026524  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
026276  000403                        BR      $ATYC
026300  112737  000001  026526  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
026306                          $ATYC:
026306  010046                        MOV     R0,-(SP)        ;;PUSH R0 ON STACK
026310  010146                        MOV     R1,-(SP)        ;;PUSH R1 ON STACK
026312  105737  026524                TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
026316  001450                        BEQ     5$              ;;IF NOT:  BR
026320  122737  000001  001210        CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
026326  001031                        BNE     3$              ;;IF NOT:  BR
026330  132737  000100  001211        BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
026336  001425                        BEQ     3$              ;;IF NOT:  BR
026340  017600  000004                MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
026344  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
026352  005737  001170          1$:   TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
026356  001375                        BNE     1$              ;;IF NOT:  WAIT
026360  010037  001204                MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
026364  105720                  2$:   TSTB    (R0)+           ;;FIND END OF MESSAGE
026366  001376                        BNE     2$
026370  163700  001204                SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
026374  006200                        ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
026376  010037  001206                MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
026402  012737  000004  001170        MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
026410  000413                        BR      5$
026412  017637  000004  026436  3$:   MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
026420  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
026426  013746  177776                MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
026432  004737  025136                JSR     PC,$TYPE        ;;CALL TYPE MACRO
026436  000000                  4$:   .WORD   0
026440                          5$:
026440  105737  026526          10$:  TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
026444  001416                        BEQ     12$             ;;IF NOT:  BR
026446  005737  001210                TST     $ENV            ;;RUNNING UNDER APT?
026452  001413                        BEQ     12$             ;;IF NOT:  BR
026454  005737  001170          11$:  TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
026460  001375                        BNE     11$             ;;IF NOT:  WAIT
026462  017637  000004  001172        MOV     @4(SP),$FATAL   ;;GET ERROR #
026470  062766  000002  000004        ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
026476  005237  001170                INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
026502  105037  026526          12$:  CLRB    $FFLG           ;;CLEAR FATAL FLAG
026506  105037  026525                CLRB    $LFLG           ;;CLEAR LOG FLAG
026512  105037  026524                CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
026516  012601                        MOV     (SP)+,R1        ;;POP STACK INTO R1
026520  012600                        MOV     (SP)+,R0        ;;POP STACK INTO R0
026522  000207                        RTS     PC              ;;RETURN
026524     000          $MFLG:  .BYTE   0               ;;MESSG. FLAG
026525     000          $LFLG:  .BYTE   0               ;;LOG FLAG
026526     000          $FFLG:  .BYTE   0               ;;FATAL FLAG
026530                          .EVEN
000200                  APTSIZE=200
000001                  APTENV=001
000100                  APTSPOOL=100
```

```
 4667            000040                  APTCSUP=040
 4668                                    .SBTTL  ERROR HANDLER ROUTINE
 4669
 4670                                    ;;**********************************************************************
 4671                                    ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
 4672                                    ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
 4673                                    ;*AND GO TO MYTYPE ON ERROR
 4674                                    ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
 4675                                    ;*SW15=1         HALT ON ERROR
 4676                                    ;*SW13=1         INHIBIT ERROR TYPEOUTS
 4677                                    ;*SW10=1         BELL ON ERROR
 4678                                    ;*CALL
 4679                                    ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER
 4680
 4681  026530                            $ERROR:
 4682  026530  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4683  026532  105237  001103            7$:   INCB    $ERFLG          ;;SET THE ERROR FLAG
 4684  026536  001775                           BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
 4685  026540  013777  001102  152374          MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
 4686  026546  032777  002000  152364          BIT     #BIT10,@SWR     ;;BELL ON ERROR?
 4687  026554  001402                           BEQ     1$              ;;NO - SKIP
 4688  026556  104401  001160                   TYPE    ,$BELL          ;;RING BELL
 4689  026562  005237  001112            1$:   INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
 4690  026566  011637  001116                   MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
 4691  026572  162737  000002  001116          SUB     #2,$ERRPC
 4692  026600  117737  152312  001114          MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
 4693  026606  032777  020000  152324          BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
 4694  026614  001004                           BNE     20$             ;;SKIP TYPEOUTS
 4695  026616  004737  027044                   JSR     PC,MYTYPE       ;;GO TO USER ERROR ROUTINE
 4696  026622  104401  001165                   TYPE    ,$CRLF
 4697  026626                            20$:
 4698  026626  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
 4699  026634  001007                           BNE     2$              ;;NO,SKIP APT ERROR REPORT
 4700  026636  113737  001114  026650          MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
 4701  026644  004737  026300                   JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
 4702  026650     000                    21$:   .BYTE   0
 4703  026651     000                           .BYTE   0
 4704  026652  000777                    22$:   BR      22$             ;;APT ERROR LOOP
 4705  026654  005777  152260            2$:   TST     @SWR            ;;HALT ON ERROR
 4706  026660  100002                           BPL     3$              ;;SKIP IF CONTINUE
 4707  026662  000000                           HALT                    ;;HALT ON ERROR!
 4708  026664  104407                           CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
 4709  026666                            3$:
 4710  026666  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.

026670                  $ERRTYP:
026670  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
026674  010046                  MOV     R0,-(SP)        ;;SAVE R0
026676  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
026700  153700  001114          BISB    @#$ITEMB,R0
026704  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                        ;;TYPE THE PC OF THE ERROR
026706  013746  001116          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                        ;;ERROR ADDRESS
026712  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
026714  000445                  BR      10$             ;;GET OUT
026716  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
026720  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
026722  006300                  ASL     R0
026724  006300                  ASL     R0
026726  062700  001250          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
026732  012037  026742          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
026736  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
026740  104401                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
026742  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
026744  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
026750  012037  026760  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
026754  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
026756  104401                  TYPE                    ;;TYPE THE "DATA HEADER"
026760  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
026762  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
026766  010146          5$:     MOV     R1,-(SP)        ;;SAVE R1
026770  012001                  MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
026772  001415                  BEQ     9$              ;;BR IF NO DATA TO BE TYPED
026774  012000                  MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
026776  105720          6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
027000  001003                  BNE     7$              ;;BR IF DECIMAL
027002  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
027004  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
027006  000402                  BR      8$
027010                  7$:
027010  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
027012  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
027014  005711          8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
027016  001403                  BEQ     9$              ;;BR IF NO
027020  104401  027040          TYPE    ,11$            ;;TYPE TWO(2) SPACES
027024  000764                  BR      6$              ;;LOOP

027026  012601          9$:     MOV     (SP)+,R1        ;;RESTORE R1
027030  012600          10$:    MOV     (SP)+,R0        ;;RESTORE R0
027032  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
027036  000207                  RTS     PC              ;;RETURN
027040  020040  000     11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
027044                          .EVEN
```

```
 4767
 4768
 4769
 4770  027044  113737  001114  027076  MYTYPE: MOVB    $ITEMB,ERRNUM   ;FOR ERROR TYPEOUT
 4771  027052  013737  014244  001174          MOV     SRXSA,$TESTN    ;FOR APT
 4772                                                                  ;APT PRINTS $TESTN & $FATAL IN THAT ORDER.
 4773                                                                  ;$TESTN WILL BE FLOPPY RXSA
 4774                                                                  ;$FATAL IS ERR #
 4775  027060  004737  026670                  JSR     PC,$ERRTYP      ;TYPE ERR MSG
 4776  027064  005237  027100                  INC     TERR
 4777  027070  005237  027104                  INC     PERR
 4778  027074  000207                          RTS     PC
 4779
 4780  027076  000000                  ERRNUM: 0                       ;FOR ERR TYPEOUT
 4781
 4782  027100  000000                  TERR:   0                       ;TOTAL ERROR COUNT
 4783  027102  000000                  TSERR:  0                       ;TOTAL SOFT ERR COUNT
 4784  027104  000000                  PERR:   0                       ;PASS ERR COUNT
 4785  027106  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

027110                          $TYPDS:
027110  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
027112  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
027114  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
027116  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
027120  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
027122  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
027126  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
027132  100004                          BPL     1$              ;;BR IF INPUT IS POS.
027134  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
027136  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
027144  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
027146  012703  027324                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
027152  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
027156  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
027160  016001  027314                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
027164  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
027166  002402                          BLT     4$              ;;BR IF DONE
027170  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
027172  000774                          BR      3$
027174  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
027176  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
027200  001002                          BNE     5$              ;;FALL THROUGH IF 0
027202  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
027204  100407                          BMI     7$              ;;BR IF YES
027206  106316                  5$:     ASLB    (SP)            ;;MSD?
027210  103003                          BCC     6$              ;;BR IF NO
027212  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
027220  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
027224  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
027230  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
027232  005720                          TST     (R0)+           ;;JUST INCREMENTING
027234  020027  000010                  CMP     R0,#10          ;;CHECK THE TABLE INDEX
027240  002746                          BLT     2$              ;;GO DO THE NEXT DIGIT
027242  003002                          BGT     8$              ;;GO TO EXIT
027244  010502                          MOV     R5,R2           ;;GET THE LSD
027246  000764                          BR      6$              ;;GO CHANGE TO ASCII
027250  105726                  8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
027252  100003                          BPL     9$              ;;BR IF NO
027254  116663  177777  177776          MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
027262  105013                  9$:     CLRB    (R3)            ;;SET THE TERMINATOR
027264  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
027266  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
027270  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
 4842  027272  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
 4843  027274  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
 4844  027276  104401  027324                  TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
 4845  027302  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 4846  027310  012616                          MOV     (SP)+,(SP)
 4847  027312  000002                          RTI                     ;;RETURN TO USER
 4848  027314  023420                $DTBL:    10000.
 4849  027316  001750                          1000.
 4850  027320  000144                          100.
 4851  027322  000012                          10.
 4852  027324  000004                $DBLK:    .BLKW   4
 4853                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
 4854
 4855                                ;;**************************************************************
 4856                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
 4857                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
 4858                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
 4859                                ;*CALL:
 4860                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 4861                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
 4862                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
 4863                                ;*      .BYTE   M               ;;M=1 OR 0
 4864                                ;*                                      ;;1=TYPE LEADING ZEROS
 4865                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
 4866                                ;*
 4867                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
 4868                                ;*$TYPOS OR $TYPOC
 4869                                ;*CALL:
 4870                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 4871                                ;*      TYPON                   ;;CALL FOR TYPEOUT
 4872                                ;*
 4873                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
 4874                                ;*CALL:
 4875                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
 4876                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
 4877
 4878  027334  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
 4879  027340  116637  000001  027557        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
 4880  027346  112637  027561              MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
 4881  027352  062716  000002              ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
 4882  027356  000406                      BR      $TYPON
 4883  027360  112737  000001  027557  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
 4884  027366  112737  000006  027561        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
 4885  027374  112737  000005  027556  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 4886  027402  010346                      MOV     R3,-(SP)        ;;SAVE R3
 4887  027404  010446                      MOV     R4,-(SP)        ;;SAVE R4
 4888  027406  010546                      MOV     R5,-(SP)        ;;SAVE R5
 4889  027410  113704  027561              MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 4890  027414  005404                      NEG     R4
 4891  027416  062704  000006              ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 4892  027422  110437  027560              MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 4893  027426  113704  027557              MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 4894  027432  016605  000012              MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 4895  027436  005003                      CLR     R3              ;;CLEAR THE OUTPUT WORD
 4896  027440  006105                1$:   ROL     R5              ;;ROTATE MSB INTO "C"
 4897  027442  000404                      BR      3$              ;;GO DO MSB
```

```
4898  027444  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
4899  027446  006105                          ROL     R5
4900  027450  006105                          ROL     R5
4901  027452  010503                          MOV     R5,R3
4902  027454  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
4903  027456  105337  027560                  DECB    $OMODE          ;;TYPE THIS DIGIT?
4904  027462  100016                          BPL     7$              ;;BR IF NO
4905  027464  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
4906  027470  001002                          BNE     4$              ;;TEST FOR 0
4907  027472  005704                          TST     R4              ;;SUPPRESS THIS 0?
4908  027474  001403                          BEQ     5$              ;;BR IF YES
4909  027476  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
4910  027500  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
4911  027504  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
4912  027510  110337  027554                  MOVB    R3,8$           ;;SAVE FOR TYPING
4913  027514  104401  027554                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
4914  027520  105337  027556          7$:     DECB    $OCNT           ;;COUNT BY 1
4915  027524  003347                          BGT     2$              ;;BR IF MORE TO DO
4916  027526  002402                          BLT     6$              ;;BR IF DONE
4917  027530  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
4918  027532  000744                          BR      2$              ;;GO DO THE LAST DIGIT
4919  027534  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
4920  027536  012604                          MOV     (SP)+,R4        ;;RESTORE R4
4921  027540  012603                          MOV     (SP)+,R3        ;;RESTORE R3
4922  027542  016666  000002  000004         MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
4923  027550  012616                          MOV     (SP)+,(SP)
4924  027552  000002                          RTI                     ;;RETURN
4925  027554     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
4926  027555     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
4927  027556     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
4928  027557     000          $ZFILL: .BYTE   0               ;;ZERO FILL SWITCH
4929  027560  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.

027562  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
027564  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
027570  005740                          TST     -(R0)           ;;BACKUP BY 2
027572  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
027574  006300                          ASL     R0              ;;POSITION FOR INDEXING
027576  016000  027616                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
027602  000200                          RTS     R0              ;;GO TO ROUTINE


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

027604  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
027606  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
027614  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
027616  027604                  $TRPAD: .WORD   $TRAP2
027620  025136                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
027622  027360                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
027624  027334                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
027626  027374                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
027630  027110                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

027632  025470                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

027634  025420                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
027636  025742                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
027640  026062                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
027642  012152                          GT$DEV  ;;CALL=GTDEVM   TRAP+12(104412) ^C WILL OPEN UP $DEVM

027644  000040                          .BLKW   32.             ;STACK FOR "EIS-FIS" TESTS

027744                          LSTAD:
                                        ;MAX ADDR FOR 8K WITH APT = 37400
                                        ;MAX ADDR FOR 8K NO APT BUT TO SAVE ABS LOADER = 37500
        003334                  .END    START
```

```
ABASE = 176500            691#  1171  1212
ACDW1 = 000000           1171
ACDW2 = 000000           1171
ACOM    020224           2194  2914  4377#
ACPUOP= 000000           1171  1186
ADDR    004546           1741* 1759* 1764* 1792* 1827# 4377
ADDW0 = 000000           1171
ADDW1 = 000000           1171
ADDW10= 000000           1171
ADDW11= 000000           1171
ADDW12= 000000           1171
ADDW13= 000000           1171
ADDW14= 000000           1171
ADDW15= 000000           1171
ADDW2 = 000000           1171
ADDW3 = 000000           1171
ADDW4 = 000000           1171
ADDW5 = 000000           1171
ADDW6 = 000000           1171
ADDW7 = 000000           1171
ADDW8 = 000000           1171
ADDW9 = 000000           1171
ADEVCT= 000000           1171  1177
ADEVM = 000017            691# 1171  1213
AENV  = 000000           1171  1182
AENVM = 000000           1171  1183
AFATAL= 000000           1171  1174
AMADR1= 000000           1171  1199
AMADR2= 000000           1171  1203
AMADR3= 000000           1171  1206
AMADR4= 000000           1171  1209
AMAMS1= 000000           1171  1193
AMAMS2= 000000           1171  1201
AMAMS3= 000000           1171  1204
AMAMS4= 000000           1171  1207
AMOD  = 030000            905# 2171  2173  2935
AMRB  = 176612            864# 2182  3420
AMRC  = 176610            863# 1659* 1660* 1690* 2174* 2183* 2207* 2208* 2936* 2937*
AMRSRV  013656           2178  3420#
AMRVEC= 000330            837# 2177
AMSGAD= 000000           1171  1179
AMSGLG= 000000           1171  1180
AMSGTY= 000000           1171  1173
AMTST   006160           2109  2114  2158  2166#
AMTYP1= 000000           1171  1194
AMTYP2= 000000           1171  1202
AMTYP3= 000000           1171  1205
AMTYP4= 000000           1171  1208
AMXB  = 176616            866# 3417*
AMXC  = 176614            865# 2184* 3416* 3432*
AMXSRV  013640           2179  3416#
AMXVEC= 000334            838#
ANS1    005420           2019* 2024  2033#
ANS2    005422           2020* 2027  2034#
APASS = 000000           1171  1176
APRIOR= 000000           1171
```

```
APTCSU= 000040          4408    4667#
APTENV= 000001          4401    4623    4665#   4698
APTSIZ= 000200          1622    4664#
APTSPO= 000100          4403    4625    4666#
ASHC1   004732          1882    1888#
ASHC2   005002          1904    1910#
ASHC3   005054          1926    1931#
ASHC4   005074          1935    1942#
ASH1    004602          1836    1841#   2032
ASH2    004642          1853    1858#
ASH3    004664          1862    1867#
ASH4    004704          1871    1876#
ASWREG= 000000          1171    1184
ATESTN= 000000          1171    1175
AUNIT = 000000          1171    1178
AUSWR = 000000           691#   1171    1185
AVECT1= 000300           691#   1171    1210
AVECT2= 000000          1171    1211
BITHLD  012716          3154*   3158    3172#
BIT0  = 000001           810#    976    1072    2213
BIT00 = 000001           800#    810
BIT01 = 000002           799#    809
BIT02 = 000004           798#    808
BIT03 = 000010           797#    807
BIT04 = 000020           796#    806
BIT05 = 000040           795#    805
BIT06 = 000100           794#    804
BIT07 = 000200           793#    803
BIT08 = 000400           792#    802
BIT09 = 001000           791#    801
BIT1  = 000002           809#    943     963     999    1071    2247
BIT10 = 002000           790#    956     990    2107    4686
BIT11 = 004000           789#    989    1030    2166
BIT12 = 010000           788#    955     970     988    1005    1029    1804    2037    2060    2094    2157    2192    2210
                        2235    2269    2303    3791    3805    3824    4152    4166    4186
BIT13 = 020000           787#    954     969     987    2244    4693
BIT14 = 040000           786#    953     968     986    1004    1012    2278
BIT15 = 100000           785#    952     967     985    1003    1028    1047    2069    3912    4091
BIT2  = 000004           808#    962     998    1070    2281
BIT3  = 000010           807#    961     997    1036    1069    2120    2169    2928
BIT4  = 000020           806#    996    1035    1050    1051    1068
BIT5  = 000040           805#    960     995    1034    1067
BIT6  = 000100           804#    959     975     994    1033    1049    1835    2893
BIT7  = 000200           803#    958     974     993    1032    1048    1066    2047
BIT8  = 000400           802#    992    1031    2319    2419    2457    2518    2568    2592    2637    2703    2740
BIT9  = 001000           801#    957     991    2321    2438    2465    2543    2573    2666
BOTMON  004510          1768    1785    1798    1814#
BPTVEC= 000014           817#
BREAK = 000001           976#
B110  = 001000           912#
B1200 = 003400           917#
B134  = 001400           913#
B150  = 002000           914#
B1800 = 004000           918#
B19200= 007400           925#
B2000 = 004400           919#
```

```
B2400 = 005000            920#  2215  2217  2249  2251  2283  2285
B300  = 002400            915#
B3600 = 005400            921#
B4800 = 006000            922#
B50   = 000000            910#
B600  = 003000            916#
B7200 = 006400            923#
B75   = 000400            911#
B9600 = 007000            924#  2074  2171  2173  2935
CDET  = 010000            955#   988#
CHAR5 = 000000            936#
CHAR6 = 000004            937#
CHAR7 = 000010            938#
CHAR8 = 000014            939#  2074  2171  2173  2215  2217  2249  2251  2283  2285  2935
CHR5  = 000000           1016#
CHR6  = 002000           1017#
CHR7  = 004000           1018#
CHR8  = 006000           1019#  2123
CKSWR = 104407           2988   4682  4708  4969#
CKOBAD  013010           2531   3219#
CK1BAD  013122           2556   3267#
CLK   = 177546            858#  1684*  2052*
CLKTST  005446           2047#
CLKVEC= 000100            835#  1575*  1576*  2049*  2050*
CLOCK   020300           2062   2910   4377#
CLOPT   020136           2881   4377#
CLPRES  012014           2201   2877*  2880*  2946#
CLRBAD  012720           1698   3179#
CLT1    020156           2237   2915   4377#
CLT2    020167           2271   2916   4377#
CLT3    020200           2305   2917   4377#
CL1TST  006320           2167   2190   2193   2201#
CL2TST  006510           2211   2233   2236   2244#
CL3TST  006642           2245   2267   2270   2278#
COMCHR  013734           2146*  2151   2181*  2187  3417  3421  3425  3427*  3430*  3435#  3443  3444  3467
                         3471   3473*  3476*  4377
COMHI D 013736           2135*  2136   2182*  3420*  3421  3436#  3464*  3465  3467  4377
COMPAT  021403           2807   4377#
COMP1   003272           1556#  1562*  1566*  1570*  1578*  1648  1700  2473  2805  2809*  3908  4087
COMP2   003274           1557#  1565*  1571*  1579*  1650  1702  2425  2444  2578  2810*  3910  4089
CONSOL= 010000            900#
COPDUN  021311           4282   4377#
COPFLG  003270           1555#  1569*  1580*  1645  1708  4210*
COPMSG  021210           4211   4377#
COPY    017410           1710   4210#  4284
CR    = 000015            725#  2079   3310  3323  4447  4457
CRC   = 000010           1069#  3658   4032
CRLF  = 000200            726#  1644   4377  4418  4457
CTHLD   012434           3065*  3069   3083#
CTREG   012524           3093*  3097*  3108#  3121*  3125*
CTREG1  012526           3092*  3100*  3109#  3120*  3128*
CTS   = 020000            954#   987#
CT1   = 000000            901#  2215  2217
CT2   = 050000            902#  2249  2251
CT3   = 060000            903#  2283  2285
DATCHK  020102           4266   4357#
```

```
DD     = 000040          1067#  3681
DDISP  = 177570           732#  1154  1610
DEVM     021500          2995  4377#
DF       025110          1233  1236  1239  1242  1245  1248  1251  1254  1257  1260  1263  1266  1269
                         1308  1311  1314  1317  1320  1323  1326  1329  1332  1335  1338  1341  1344
                         1347  1350  1353  1356  1362  1365  1368  1371  1374  1377  1380  1383  1410
                         1413  1416  1419  1425  1428  1431  1434  1467  1470  1473  1476  1479  1482
                         1485  1488  1491  1494  1502  1505  4377#
DF1      025116          1272  1275  1278  1281  1290  1293  1296  1299  1386  1389  1392  1395  1398
                         1401  1404  1407  1449  1458  4377#
DF2      025126          1284  1287  1302  1305  1359  1452  1461  4377#
DH1      024170          1248  1308  1314  1317  1320  1323  1326  1335  1338  1341  1502  1505  4377#
DH2      024207          1233  1236  1242  4377#
DH3      024233          1239  1245  4377#
DH4      024275          1251  1311  4377#
DH5      024326          1254  1257  1260  1263  1266  1416  1419  4377#
DH6      024363          1269  1329  1332  1344  1347  1350  1353  1356  1362  1365  1368  1371  1374
                         1377  1380  1383  1410  1413  1425  1428  1431  1434  1467  1470  1473  1476
                         1479  1482  1485  1488  1491  1494  4377#
DH7      024430          1272  1275  1278  1281  1290  1293  1296  1299  1386  1389  1392  1395  1398
                         1401  1404  1407  1449  1458  4377#
DH8      024517          1284  1287  1302  1305  1359  1452  1461  4377#
DISPLA   001142          1154#  1610*  1618*  4685*
DISPRE   000174          1081#  1618
DIV0     005172          1967  1974#
DNA    = 100000          1028#
DNAIE  = 000040          1034#
DONE   = 000200           958#   993#  1048#  2131  2326  2334  2353  2366  4328  4347
DROP     020663          2906  4377#
DRV0     020266          2911  3793  3807  3826  4377#
DRV1     020273          2912  4154  4168  4188  4377#
DSC    = 100000           985#
DSI    = 100000           952#
DSIE   = 000040           960#   995#
DSRDY  = 001000           957#   991#
DSWR   = 177570           731#  1153  1609
DTR    = 000002           963#   999#  1659  1660  1690  2124  2174  2207  2208  2936  2937
DT1      024606          1248  1308  1314  1317  1320  1323  1326  1335  1338  1341  1502  1505  4377#
DT10     024736          1272  1275  1278  1281  1386  1389  1392  1395  1449  4377#
DT11     024760          1284  1287  1359  1452  4377#
DT12     025002          1290  1293  1296  1299  1398  1401  1404  1407  1458  4377#
DT13     025024          1302  1305  1461  4377#
DT14     025046          1329  1332  1344  1347  1350  1353  1368  1374  1413  1425  1428  1431  1434
                         1467  1470  1473  1476  1479  1482  1485  1488  1491  1494  4377#
DT15     025064          1416  4377#
DT16     025076          1419  4377#
DT2      024614          1233  1236  1242  4377#
DT3      024624          1239  1245  4377#
DT4      024640          1251  1311  4377#
DT5      024650          1254  4377#
DT6      024662          1257  4377#
DT7      024674          1260  4377#
DT8      024706          1263  1266  4377#
DT9      024720          1269  1356  1362  1365  1371  1377  1380  1383  1410  4377#
DUN      021011          1807  2040  2160  3797  3809  3828  4158  4170  4190  4377#
DXTST    003260          1543#  2317*  2404*  2505*  2595*  2640*  2706*  2744*  3553  3679  3741  3765  3836
```

```
                    3838    4197    4377
DXTST0  006774      2203    2279    2301    2304    2317#
DXTST1  007246      1706    2322    2383    2404#
DXTST2  007702      2478    2488    2505#
DXTST3  010252      2579    2592#
DXTST4  010426      2593    2606    2637#
DXTST5  010700      2638    2667    2675    2703#
DXTST6  011016      2704    2717    2726    2740#
DX1   = 000020      1050#   2451    2562    2671    2687    3863    3893    3915    3931    3951    3986    3999    4020
                    4029    4068    4172    4191    4239    4252
DOBAD   002764      1523#   3179    3181    3197    3204    3226    3227
DOBUF   002564      1522#   2358    2371    2381    3698    3719    4236    4358
DOCDAT  015140      3699    3714#
DOCERR  002554      1518#   2411*   2511*   2597*   2617*   2657*   2708*   3739    3754    3760*   3764*   4377
DOCMER  002552      1517#   3605*   3714*   3724*   3732    3781*   3819*
DOCNT   002542      1513#   2512*   2570    3821*   3822
DOCRC   002562      1521#   3660*   3664*   3734    3749    3761*
DODISP  014230      2427*   2430*   2534*   2599*   2621*   2641*   2659*   2710*   2719*   2749*   3512    3513#   3533*
                    3552*   3576*   3611*   3625*   3645*   3654*   3699*   3849*   3858*   3870*   3878*
DODUN   016002      3769    3840    3849    3858    3870    3878    3888#
DOEBUF  015102      2621    3685    3697#
DOERR   002544      1514#   2410*   2510*   2596*   2616*   2656*   2707*   3570    3583    3593*   3604*   3608*   3640
                    3642*   3668    3678*   3780*   3818*   4377
DOEXP   002556      1519#   3722*   4377
DOFBUF  014246      2430    2710    3529#   3601    3611
DOINIT  015666      2599    3847#
DOINV   015764      2749    3878#
DONEXT  015644      3654    3666    3762    3836#
DONXT1  015364      3780#   3837
DONXT2  015560      3818#   3839
DOPAT   002540      1512#   2409*   2459    3810*
DORCHK  014656      3625    3638#
DOREC   002560      1520#   3723*   4377
DOREST  015712      2641    3858#
DORSEC  014622      2427    2534    2659    3606    3620#   3644    3645    3767    3789
DOSEC   002550      1516#   2412*   2529*   2619*   2654*   2746*   2772*   3224    3532    3550    3594    3596*   3598
                    3600*   3603*   3622    3718    3782    3784*   3786    3788*   3802*   4377
DOSTAT  015746      2719    3870#
DOTMO   012530      2461    3091*   3102*   3110#
DOTRK   002546      1515#   2406*   2482*   2489*   2524*   2618*   2653*   2745*   2771*   3222    3530    3548    3620
                    3716    3795    3799    3801*   4377
DOWCHK  014366      3552    3568#
DOWSEC  014312      3533    3548#
D1BAD   003234      1541#   3184    3186    3245    3252    3274    3275
D1BUF   003034      1540#   4066    4093    4264    4359
D1CDAT  016702      4067    4081#
D1CERR  003024      1536#   2416*   2515*   2684*   4111    4120    4126*   4130*   4377
D1CMER  003022      1535#   3979*   4081*   4098*   4105    4142*   4181*
D1CNT   003012      1531#   2516*   2575    4183*   4184
D1CRC   003032      1539#   4034*   4038*   4107    4116    4127*
D1DISP  014236      2446*   2449*   2559*   2669*   2686*   3515    3516#   3862*   3914*   3930*   3950*   3985*   3998*
                    4019*   4028*   4067*
D1DUN   016016      3862    3892#
D1EBUF  016644      4053    4065#
D1ERR   003014      1532#   2415*   2514*   2683*   3944    3957    3967*   3978*   3982*   4014    4016*   4042    4052*
                    4141*   4180*   4377
```

```
D1EXP   003026          1537#   4096*   4377
D1FBUF  016032          2449    3903#   3975    3985
D1INT   014232          3510    3515#
D1NEXT  017376          4028    4040    4128    4197#
D1NXT1  017116          4141#   4198
D1NXT2  017312          4180#   4199
D1PAT   003010          1530#   2414*   2467    4171*
D1RCHK  016442          3998    4012#
D1REC   003030          1538#   4097*   4377
D1REST  015730          2669    3862#
D1RSEC  016406          2446    2559    2686    3980    3993#   4018    4019    4131    4150
D1SEC   003020          1534#   2417*   2554*   2682*   3272    3906    3928    3968    3970*   3972    3974*   3977*   3995
                        4085    4143    4145*   4147    4149*   4163*   4377
D1TMO   012620          2469    3119*   3130*   3136#
D1TRK   003016          1533#   2407*   2483*   2490*   2549*   2681*   3270    3904    3926    3993    4083    4156    4160
                        4162*   4377
D1WCHK  016152          3930    3942#
D1WSEC  016116          3914    3926#
EBHLD   007240          2372*   2376*   2377    2387#   4377
EBUF  = 000003          1056#   2359    3700    4068    4341
EBUFF   020040          4235    4263    4339#
EIPRES  012012          1833    2851*   2857*   2945#
EISCNT  005424          1832*   2030*   2035#
EISDON  005426          2031    2037#
EISFIS  004552          1805    1808    1832#
EISTXT  020307          2859    2909    4377#
EMTVEC= 000030           820#   1601*   1602*
EM1     021511          1233    1236    1242    4377#
EM10    022024          1269    4377#
EM11    022051          1272    1275    4377#
EM13    022067          1278    1425    4377#
EM14    022120          1281    4377#
EM15    022151          1284    4377#
EM16    022202          1287    4377#
EM17    022233          1290    1293    1428    4377#
EM19    022251          1296    1434    4377#
EM2     021530          1239    1245    4377#
EM20    022302          1299    4377#
EM21    022333          1302    4377#
EM22    022364          1305    4377#
EM23    022415          1308    4377#
EM24    022426          1311    4377#
EM25    022443          1314    4377#
EM26    022460          1317    4377#
EM27    022475          1320    4377#
EM28    022512          1323    4377#
EM29    022531          1326    4377#
EM3     021552          1248    4377#
EM30    022551          1329    1344    1347    1350    1353    1368    1374    1476    1482    4377#
EM31    022562          1332    1479    1485    1488    4377#
EM32    022573          1335    4377#
EM33    022636          1338    4377#
EM34    022677          1341    4377#
EM35    022736          1356    4377#
EM36    023003          1359    4377#
EM37    023052          1362    4377#
```

```
EM38    023076            1365    4377#
EM39    023144            1371    4377#
EM4     021616            1251    4377#
EM40    023174            1377    4377#
EM41    023237            1380    4377#
EM42    023273            1383    4377#
EM43    023314            1386    1389    4377#
EM45    023346            1392    1395    4377#
EM47    023377            1398    1401    4377#
EM49    023431            1404    1407    4377#
EM5     021641            1254    4377#
EM51    023462            1410    4377#
EM52    023506            1413    1467    1470    1473    1491    1494    4377#
EM53    023532            1416    1419    4377#
EM54    023601            1431    4377#
EM55    023616            1440    4377#
EM56    023653            1443    4377#
EM57    023710            1449    4377#
EM58    023731            1452    4377#
EM59    023752            1455    4377#
EM6     021667            1257    4377#
EM60    024010            1458    4377#
EM61    024031            1461    4377#
EM62    024052            1464    4377#
EM63    024110            1502    4377#
EM64    024145            1505    4377#
EM7     021715            1260    4377#
EM8     021743            1263    4377#
EM9     021774            1266    4377#
ENDPAS  021017            2786    4377#
ENVPAR= 001400            1023#
EOP     011176            2385    2475    2580    2756    2785#
EPAR  = 000060             931#
ERFLG   017722            4216*   4227*   4228    4234*   4243*   4244    4250*   4256*   4257    4288#
ERR   = 100000             967#   1003#   1047#
ERRNUM  027076            4377    4770*   4780#
ERRVFC= 000004             813#   1607    1608*   1619*   1725*   1726*   1801*   1802*   2821*   2822*   2884*   2885*
EXAMKB  012136            1723    2986#   3071    3096    3124    3160
EXP0    007242            2388#   4377
EXP1    007244            2389#   4377
FBUF  = 000001            1055#   2345    3534    3915
FIN     003262            1550#   2431*   2450*   2536*   2561*   2601*   2625*   2642*   2658*   2670*   2685*   2712*   2721*
                          2751*   3094    3122    3798*   3820*   3888*   3892*   4159*   4182*
FIS     005236            1986    1996#
FISEIS  020757            2039    4377#
FLGHLD  012432            3064*   3069    3082#
FRERR = 020000             969#
GETSEC  017730            4214    4296#
GNS   = ****** U          1643    4961    4962    4963    4964    4965    4967    4969    4970    4971    4972
GTDEVM= 104412            1666    4972#
GTSWR = 104406            1638    4967#
GT$DEV  012152            2995#   4498    4584    4972
HFDUP = 000010            1036#
HILIM   012130            2522*   2527*   2547*   2552*   2975    2980#
HINUM   012132            2957    2965    2970*   2982#
HLD42   004550            1588*   1814    1828#
```

```
HT     = 000011            723#  4416  4457
ID     = 000001           1072#  2612
IE     = 000100            959#   975#   994#  1033#  1049#  1684  2052  2081  2092  2147  2148  2183  2184
                          2226   2227   2260   2261   2294   2295  2432  2451  2537  2562  2602  2624  2643
                          2660   2671   2687   2713   2722   2752  3336  3352  3363  3379  3389  3405  3416
                          3432   3442   3450   3477   3481   3534  3555  3557  3577  3612  3626  3646  3655
                          3700   3811   3829   3850   3859   3883  3871  3879  3889  3893  3915  3931  3951
                          3986   3999   4020   4029   4068   4172  4191
INITAL= 000011           1059#  2330   3850
INTFLG  013204           1737*  1739   1755*  1757   2051*  2055  2829*  2833  2852*  2855  2867*  2870  3294*
                         3296#
INTSRV  013176           1725   2049   2821   2823   3294#
INVADR= 000002           1071#  2758
INVCT   011174           2743*  2767   2770*  2778#
IOTVEC= 000020            818#
LDATA  = 000004          1070#
LDOBAD  012752           3195#  3572   3649
LD1BAD  013064           3243#  3946   4023
LF     = 000012            724#  3312   3319   4377   4451  4457
LINCT   013334           2078*  2085   3324*  3329#
LOLIM   012126           2521*  2526*  2546*  2551*  2973  2979#
LONUM   012134           2956   2963   2969*  2983#
LOOP    004072           1646   1649   1651   1670   1674  1679#  2811
LSTAD   027744           1744   1753   1781   1789   1996  2021   4296  4976#
LSTCHR  014026           2116*  3447   3449*  3458#
MAINT  = 000002            943#  2171   2215   2249   2283
MAXMEM  012022           1728   2842*  2950#
MAXSEC  017716           4270   4286#  4309*
MAXSK   003266           1553#  1654*  2570   2575   2803*  3822  4184
MAXTRK  003264           1551#  1653*  1705*  2480   2484*  2487  2491*  2508*  2522  2547  2802*  3799  4160
MCLK   = 004000          1030#  2118   2122
MEM     020327           2846   4377#
MEMHLD  004542           1761*  1762   1794*  1825#  4377
MEMORY  020776           1806   4377#
MEMTST  004204           1709   1723#
MM     = 010000          1029#  2122
MR     = 000400          1031#  2117   2156
MSG1    021034           2789   4377#
MSG2    021063           2792   4377#
MSG3    021117           2795   4377#
MSG4    021154           2798   4377#
MULO    005132           1951   1957#
MYTYPE  027044           4695   4770#
NOEIS   005446           1837   2038   2041#
NOTPRE  020350           2860   2882   2932   4377#
NULL    011334           2801   2813#
ODDPAR= 001000           1024#  2123
OK      020715           4377#
OPAR   = 000020            932#  2171   2173   2935
ORERR  = 040000            968#  1004#
PARAM  = 177420            856#  2074*  2171*  2173*  2215*  2217*  2249*  2251*  2283*  2285*  2935*
PARERR= 010000             970#  1005#
PAT     004544           1752*  1756   1762   1774*  1793*  1826#  4377
PATT    020726           3808   4169   4377#
PERR    027104           1695*  2796   4777*  4784#
PIRQ   = 177772            730#
```

```
PIRQVE= 000240   824#
PORT    020251  2099    4377#
PRB   = 177516   861#    3309*
PRCHR   013332  2079*    3309    3310    3312    3314    3316*   3319*   3321*   3323*   3328#
PRINT   020240  2096    2918    2931    4377#
PRPORT  012020  2090    2097    2924*   2940*   2948#
PRPRES  012016  2071    2923*   2941*   2947#
PRS   = 177514   860#    2081*   2092*   2925    2938    3303
PRSRV   013206  2075    3303#
PRT   = 020000   904#    2074
PRTST   005540  2048    2058    2061    2069#
PRVEC = 000200   833#    2075*   2076*
PR0   = 000000   747#    1688
PR1   = 000040   748#
PR2   = 000100   749#
PR3   = 000140   750#
PR4   = 000200   751#    1587
PR5   = 000240   752#
PR6   = 000300   753#
PR7   = 000340   754#
PS    = 177776   727#     728
PSERR   027106  1696*    2799    3586*   3671*   3757*   3960*   4045*   4123*   4785#
PSW   = 177776   728#
PWRVEC= 000024   819#
RACT  = 004000   989#
RAND    012024  2523    2528    2548    2553    2956#   2974    2976
RANDOM  020741  3827    4189    4377#
RDCHR = 104410  4578    4970#
RDLIN = 104411  4971#
RDWR    020000  4222    4238    4251    4321#
RDY   = 000200   974#    1032#   1066#
REGHLD  012714  3153*   3158    3171#
RESTOR= 000017  1062#    3577    3612    3646    3655    3859    3863    3951    3986    4020    4029
RESVEC= 000010   814#    2823*   2824*   2886*   2887*
RING  = 040000   953#     986#
RNF   = 000020  1068#    3573    3587    3609    3643    3650    3672    3947    3961    3983    4017    4024    4046
RSEC  = 000007  1058#    3626    3999    4223    4252
RSTAT = 000013  1060#    3871
RTS   = 000004   962#     998#   1659    1660    1690    2124    2174    2207    2208    2936    2937
RUN     020704  2063    2101    2195    2238    2272    2306    4377#
RXCS  = 177170   874#    2325    2330*   2333    2338    2345*   2352    2359*   2365    2432*   2451*   2537*   2562*
                2602*   2608    2624*   2643*   2649    2660*   2671*   2677    2687*   2713*   2722*   2728    2752*
                2763    3494    3496    3502    3505    3509    3534*   3555*   3557*   3568    3577*   3612*   3626*
                3638    3646*   3655*   3700*   3811*   3829*   3850*   3859*   3863*   3871*   3879*   3889*   3893*
                3915*   3931*   3942    3951*   3986*   3999*   4012    4020*   4029*   4068*   4172*   4191*   4219
                4225    4241    4254    4325*   4327    4341*   4346    4377
RXDB  = 177172   875#    2346*   2360    3535*   3701    3916*   4069    4342
RXES  = 177172   876#    2612    2758    3497    3506    3681    4377
RXSA  = 177174   877#    2747*   2773*   3200    3248    3498    3507    3551*   3623*   3847*   3929*   3996*   4324*
                4377
RXSRV   014130  2422    2441    3494#
RXVEC = 000264   843#    2422*   2423*   2441*   2442*
SCOM    020211  2159    2913    4377#
SECCNT  017720  4268*   4270    4287#   4312*
SECT    017726  4218*   4273    4275*   4279*   4290#   4323
SEND  = 000020  1035#    2126    3454
```

```
SETVEC  012326          2141    2176    2219    2253    2287    3045#
SIZE    011340          1668    2821#
SMPAR = 176622           870#   2123*
SMRB  = 176622           869#   2135    3464
SMRC  = 176620           868#   2124*   2130    2147*   3477*
SMRSRV  014030          2143    3464#
SMRVEC= 000340           840#   2142
SMTST   005704          2070    2072    2088    2095    2100    2107#
SMXB  = 176626           872#   2127*   3443*
SMXC  = 176624           871#   2117*   2118*   2122*   2126*   2148*   2156*   3442*   3450*   3454*   3481*
SMXSRV  013740          2144    3442#
SMXVEC= 000344           841#
SPRS    013336          3303*   3304    3330#   4377
SREC  = 002000           956#    990#
SRSYN = 000020           996#   2124
SRXCS   014240          3496*   3505*   3519#   4377
SRXES   014242          3497*   3506*   3520#   3573    3587    3609    3643    3650    3658    3672    3947    3961    3983
                        4017    4024    4032    4046    4377
SRXSA   014244          3498*   3507*   3521#   4377    4771
STACK = 001100           718#   1582    1599
START   003334          1089    1092    1575#   4979
START1  003276          1097    1562#
START2  003304          1099    1565#
START3  003316          1095    1569#
STKLMT= 177774           729#
STSYN = 000400           992#
SWR     001140          1153#   1597    1609*   1611    1617*   1624*   1636    1804    2037    2060    2094    2157    2192
                        2235    2269    2303    3791    3805    3824    4152    4166    4186    4467    4512*   4686    4693
                        4705
SWREG   000176          1082#   1617    1636    4467    4480
SWC   = 000001           782#
SW00  = 000001           772#    782
SW01  = 000002           771#    781
SW02  = 000004           770#    780
SW03  = 000010           769#    779
SW04  = 000020           768#    778
SW05  = 000040           767#    777
SW06  = 000100           766#    776
SW07  = 000200           765#    775
SW08  = 000400           764#    774
SW09  = 001000           763#    773
SW1   = 000002           781#
SW10  = 002000           762#
SW11  = 004000           761#
SW12  = 010000           760#
SW13  = 020000           759#
SW14  = 040000           758#
SW15  = 100000           757#
SW2   = 000004           780#
SW3   = 000010           779#
SW4   = 000020           778#
SW5   = 000040           777#
SW6   = 000100           776#
SW7   = 000200           775#
SW8   = 000400           774#
SW9   = 001000           773#
```

```
SXMIT = 000010            961#    997#
SYNC1 = 040000           1012#
SYNC2 = 000000           1011#
S1      003354           1563    1579#
S2      003360           1567    1580#
S3      003364           1572    1582#
TBITVE= 000014            815#
TERR    027100           1656*   2790    4776*   4782#
TIMDO   012436           2434    2539    2604    2626    2645    2662    2715    2724    2754    3091#
TIMD1   012532           2453    2564    2673    2689    3119#
TIMER   012352           2054    2084    2150    2186    2229    2263    2297    3064#
TIMER1  012622           1681    1693    3142#
TIMER2  012634           2129    2324    2332    2351    2364    3153#   4326    4345
TKVEC = 000060            822#
TK1B  = 176502            880#    2225    3340
TK1S  = 176500            879#    2226*   2868
TK1SRV  013356           2221    3340#
TK1VEC= 000300            845#    2220
TK2B  = 176512            885#    2259    3367
TK2S  = 176510            884#    2260*
TK2SRV  013456           2255    3367#
TK2VEC= 000310            847#    2254
TK3B  = 176522            890#    2293    3393
TK3S  = 176520            889#    2294*   2872
TK3SRV  013556           2289    3393#
TK3VEC= 000320            849#    2288
TPVEC = 000064            823#
TP1B  = 176506            882#    3337*
TP1S  = 176504            881#    2227*   3336*   3352*
TP1SRV  013340           2222    3336#
TP1VEC= 000304            846#
TP2B  = 176516            887#    3364*
TP2S  = 176514            886#    2261*   3363*   3379*
TP2SRV  013440           2256    3363#
TP2VEC= 000314            848#
TP3B  = 176526            892#    3390*
TP3S  = 176524            891#    2295*   3389*   3405*
TP3SRV  013540           2290    3389#
TP3VEC= 000324            850#
TRACK   017724           4217*   4278*   4289#   4321
TRAPVE= 000034            821#    1603*   1604*
TRK     020721           3794    4155    4377#
TRTVEC= 000014            816#
TSERR   027102           1657*   2793    3585*   3670*   3756*   3959*   4044*   4122*   4783#
TYPDS = 104405           2788    2791    2794    2797    2800    2845    3796    4157    4755    4965#
TYPE  = 104401           1630    1675    1806    1807    2039    2040    2062    2063    2096    2099    2101    2159    2160
                         2194    2195    2237    2238    2271    2272    2305    2306    2786    2789    2792    2795    2798
                         2801    2807    2846    2859    2860    2881    2882    2904    2906    2931    2932    2995    2998
                         3014    3033    3793    3794    3797    3807    3808    3809    3826    3827    3828    4154    4155
                         4158    4168    4169    4170    4188    4189    4190    4211    4282    4421    4478    4479    4482
                         4493    4503    4514    4533    4582    4587    4590    4594    4688    4696    4719    4736    4738
                         4741    4743    4758    4763    4844    4913    4961#
TYPOC = 104402           2997    4481    4727    4751    4962#
TYPON = 104404           4964#
TYPOS = 104403           4963#
TICHR   013434           2224*   2230    3337    3341    3345    3347*   3350*   3355#   4377
```

```
T1HLD   013436          2225* 3340* 3341  3356# 4377
T2CHR   013534          2258* 2264  3364  3368  3372  3374* 3377* 3382# 4377
T2HLD   013536          2259* 3367* 3368  3383# 4377
T3CHR   013634          2292* 2298  3390  3394  3398  3400* 3403* 3408# 4377
T3HLD   013636          2293* 3393* 3394  3409# 4377
ULITE = 040000           906#
USEL  = 000020          1051# 3509
WARNIN  020365          1675  4377#
WDDSEC= 000015          1061# 3555
WSEC  = 000005          1057# 3557  3879  3931  4239
$APTHD  001000          1113  1119#
$ASTAT= ****** U        4645  4660
$ATYC   026306          4616  4618#
$ATY1   026262          4614#
$ATY3   026270          4406  4615#
$ATY4   026300          4617# 4701
$AUTOB  001134          1150# 1640* 4475  4610
$BASE   001244          1212#
$BDADR  001122          1145#
$BDDAT  001126          1147#
$BELL   001160          1163# 4688  4711
$CHARC  025414          4423* 4433* 4440  4449* 4454#
$CKSWR  025420          4467# 4969
$CMTAG  001100          1133# 1594  1595  1605
$CM3  = 000000          1163#
$CNTLC  026220          4493  4582  4603#
$CNTLG  026232          4478  4605#
$CNTLU  026225          4503  4604#
$CPUOP  001216          1186#
$CRLF   001165          1165# 3014  4422  4457  4514  4603  4696  4711  4719  4738  4743  4763
$DBLK   027324          4810  4844  4852#
$DEVCT  001200          1177# 3074* 3099* 3127* 3163* 3325* 3351* 3378* 3404* 3431* 3451* 3455* 3478*
$DEVM   001246          1213# 1671  1673  1835  2047  2069  2107  2120  2166  2169  2205  2210  2213
                        2244  2247  2278  2281  2319  2321  2419  2438  2457  2465  2518  2543  2568
                        2573  2592  2637  2666  2703  2740  2895  2928  2996  3012*
$DTBL   027314          4813  4848#
$ENV    001210          1182# 1589  1634  1664  1669  2111  2986  4401  4623  4647  4698
$ENVM   001211          1183# 1622  4403  4408  4625
$ERFLG  001103          1136# 4683* 4711
$ERMAX  001115          1142#
$ERROR  026530          1601  4681#
$ERRPC  001116          1143# 4377  4690* 4691* 4692  4711  4725
$ERRTB  001250          1230# 4733
$ERRTY  026670          4718# 4775
$ERTTL  001112          1140# 4689* 4711
$ETABL  001210          1181#
$ETEND  001250          1125  1214#
$FATAL  001172          1174# 4651*
$FFLG   026526          4614* 4617* 4645  4654* 4662#
$FILLC  001156          1161# 4426  4457
$FILLS  001155          1160# 4457
$GDADR  001120          1144#
$GDDAT  001124          1146#
$GTSWR  025470          4479# 4967
$HD   = 000000           702
$HIBTS  001000          1120#
```

```
$ICNT   001104          1137#
$INTAG  001135          1151#   4495    4515    4610
$ITEMB  001114          1141#   4692*   4700    4711    4722    4770
$LF     001166          1166#   4457    4594    4603    4711
$LFLG   026525          4655*   4661#
$LPADR  001106          1138#
$LPERR  001110          1139#
$LSTCN= 177777           691#
$LSTIN= 000000           691#
$LSTST= 177777           691#
$LSTTA= 000000           691#
$MADR1  001222          1199#
$MADR2  001226          1203#
$MADR3  001232          1206#
$MADR4  001236          1209#
$MAIL   001170          1121    1125    1172#   1621    1634    4401    4698
$MAMS1  001220          1193#
$MAMS2  001224          1201#
$MAMS3  001230          1204#
$MAMS4  001234          1207#
$MBADR  001002          1121#
$MFLG   026524          4615*   4621    4656*   4660#
$MNEW   026250          2998    4482    4608#
$MSGAD  001204          1179#   4631*   4634
$MSGLG  001206          1180#   4636*
$MSGTY  001170          1173#   4629    4637*   4649    4653*
$MSWR   026237          4479    4606#
$MTYP1  001221          1194#
$MTYP2  001225          1202#
$MTYP3  001231          1205#
$MTYP4  001235          1208#
$NULL   001154          1159#   4428    4457
$OCNT   027556          4885*   4914*   4927#
$OMODE  027560          4880*   4884*   4889    4892*   4903*   4929#
$PASS   001176          1176#   1621*   2113    2477    2506    2785*   2787
$PASTM  001006          1123#
$QUES   001164          1164#   3033    4457    4533    4587    4603    4711
$RDCHR  025742          4546#   4970
$RDDEC= ****** U        4972
$RDLIN  026062          4574#   4971
$RDOCT= ****** U        4972
$RDSZ = 000010          4567#
$R2A  = ****** U        4972
$SAVRE= ****** U        4972
$SETUP= 000106          1560#   1600    1601    1603    1605    1630    1631    4462    4610    4682    4708    4710
$STUP = 177777          1560#
$SWR  = 132000           691#    702    1163    1605    4674    4675    4676    4677    4678    4686    4693    4705    4709
                        4711
$SWREG  001212          1184#   1624
$TESTN  001174          1175#   4771*
$TKB    001146          1156#   3005    4460    4471    4488    4550    4556
$TKS    001144          1155#   1583*   3003    4460    4469    4485    4497*   4517*   4548    4554
$TN   = 000001           691#    702
$TPB    001152          1158#   4446*   4457
$TPFLG  001157          1162#   4395    4457
$TPS    001150          1157#   4444    4457
```

```
$TRAP   027562            1603    4938#
$TRAP2  027604            4949#   4960
$TRP  = 000013            4953#   4962#   4963#   4964#   4965#   4966#   4967    4968#   4969    4970#   4971#   4972#   4973#
$TRPAD  027616            4943    4960#
$TSTM   001004            1122#
$TSTNM  001102            1135#   4685    4711
$TTYIN  026210            4575    4576    4598    4602#
$TYPBN= ****** U          4966
$TYPDS  027110            4798#   4965
$TYPE   025136            4395#   4642    4953    4961
$TYPEC  025350            3017    4425    4432    4439    4444#   4445    4519
$TYPEX  025416            4450    4452    4455#
$TYPOC  027360            4883#   4962
$TYPON  027374            4882    4885#   4964
$TYPOS  027334            4878#   4963
$UNIT   001202            1178#
$UNITM  001010            1124#
$USWR   001214            1185#
$VECT1  001240            1210#
$VECT2  001242            1211#
$OFILL  027557            4879*   4883*   4893    4928#
$40CAT= ****** U          4695
.     = 027744            1080#   1084#   1088#   1091#   1094#   1096#   1098#   1103#   1109    1110#   1112#   1114#   1132#
                          1167    1522#   1523#   1540#   1541#   1598    2729    2776    2814#   3077    3103    3131    3144
                          3166    4220    4232    4248    4261    4364    4457    4460    4602#   4603    4610#   4663#   4711
                          4766#   4852#   4974#
.$ASTA= ****** U          4615    4618
.$X   = 001000            1109#   1114
```

```
COMMEN    825#
ENDCOM    825#
ERROR     719#  1742  1760  1765  1795  1851  1854  1863  1872  1880  1883  1899  1902  1905  1921
          1924  1927  1936  1947  1952  1963  1968  1982  1987  2018  2023  2026  2029  2057  2087
          2132  2138  2153  2189  2232  2266  2300  2327  2335  2340  2354  2367  2374  2379  2435
          2454  2540  2565  2605  2610  2614  2627  2646  2651  2663  2674  2679  2690  2716  2725
          2730  2755  2760  2765  3206  3254  3306  3343  3370  3396  3423  3469  3499  3575  3580
          3589  3591  3652  3661  3665  3674  3676  3683  3736  3743  3746  3751  3758  3949  3954
          3963  3965  4026  4035  4039  4048  4050  4109  4113  4118  4124  4230  4246  4259  4329
          4348  4362
ESCAPE    825#
GETPRI    825#
GETSWR    825#  1631#
MULT      825#
NEWTST    825#
POP       825#  4657  4658  4839
PUSH      825#  4618  4620  4641  4798
REPORT    691#   825#
SCOPE     720#
SETPRI    825#
SETTRA   4953#  4962  4963  4964  4965  4967  4969  4970  4971  4972
SETUP     825#  1592
SKIP      825#
SLASH     825#
SPACE     825#
STARS     825#   829   831   852   854   894   896   946   948   979   981  1041  1043  1076  1078
         1106  1108  1115  1128  1167  1170  1508  1510  1526  1528  1713  1721  1732  1734  1747
         1749  1777  1779  1810  1812  1829  1831  1838  1840  1855  1857  1864  1866  1873  1875
         1885  1887  1907  1909  1928  1930  1939  1941  1953  1955  1971  1973  1988  1995  2043
         2045  2065  2067  2103  2105  2162  2164  2197  2199  2240  2242  2274  2276  2308  2315
         2391  2402  2495  2503  2583  2590  2630  2635  2695  2701  2732  2738  2781  2783  2817
         2819  2826  2828  2848  2850  2862  2864  2889  2891  2919  2921  2952  2954  2991  2993
         3036  3043  3058  3062  3085  3089  3113  3117  3138  3140  3147  3151  3175  3177  3191
         3193  3214  3217  3239  3241  3262  3265  3289  3292  3299  3301  3332  3334  3359  3361
         3385  3387  3412  3414  3438  3440  3485  3492  3524  3527  3544  3546  3560  3566  3616
         3618  3630  3636  3692  3695  3707  3712  3773  3778  3814  3816  3843  3845  3854  3856
         3866  3868  3874  3876  3883  3886  3898  3901  3922  3924  3934  3940  3989  3991  4004
         4010  4060  4063  4074  4079  4134  4139  4176  4178  4202  4208  4292  4294  4316  4319
         4334  4337  4353  4355  4373  4375  4380  4459  4462  4538  4567  4613  4670  4713  4788
         4855  4932
SWRSU     825#  1605#
TRMTRP   4953#
TYPBIN    825#
TYPDEC    825#  2787  2790  2793  2796  2799  2843  3795  4156  4753
TYPNAM    825#  1626
TYPNUM    825#
TYPOCS    825#
TYPOCT    825#  2996  4480  4725  4750
TYPTXT    825#
$$CMRE   1126#
$$CMTM   1126#
$$ESCA    825#
$$NEWT    825#
$$SET    4953#  4962  4963  4964  4965  4967  4969  4970  4971  4972
$$SETM   1621#
$$SKIP    825#
```

```
.EQUAT    691#     715
.HEADE    691#     692
.SETUP    691#    1560
.SWRHI    691#
.$ACTI    691#
.$APTB    691#    1168#
.$APTH    691#    1104
.$APTY    691#    4611
.$CATC    691#
.$CMTA    691#    1126
.$EOP     691#
.$ERRO    691#    4668
.$ERRT    691#    4711
.$READ    691#    4457
.$SCOP    691#
.$TRAP    691#    4930
.$TYPD    691#    4786
.$TYPE    691#    4378
.$TYPO    691#    4853


. ABS.  027744     000     OVR   RW    REL   GBL   D

 ERRORS DETECTED:  0

 Z:CVKDAC,Z:CVKDAC.SEQ/CRF/SOL-CVKDAC.P11
 RUN-TIME:  25 15 2 SECONDS
 RUN-TIME RATIO: 138/42=3.2
 CORE USED:  24K  (47 PAGES)
```