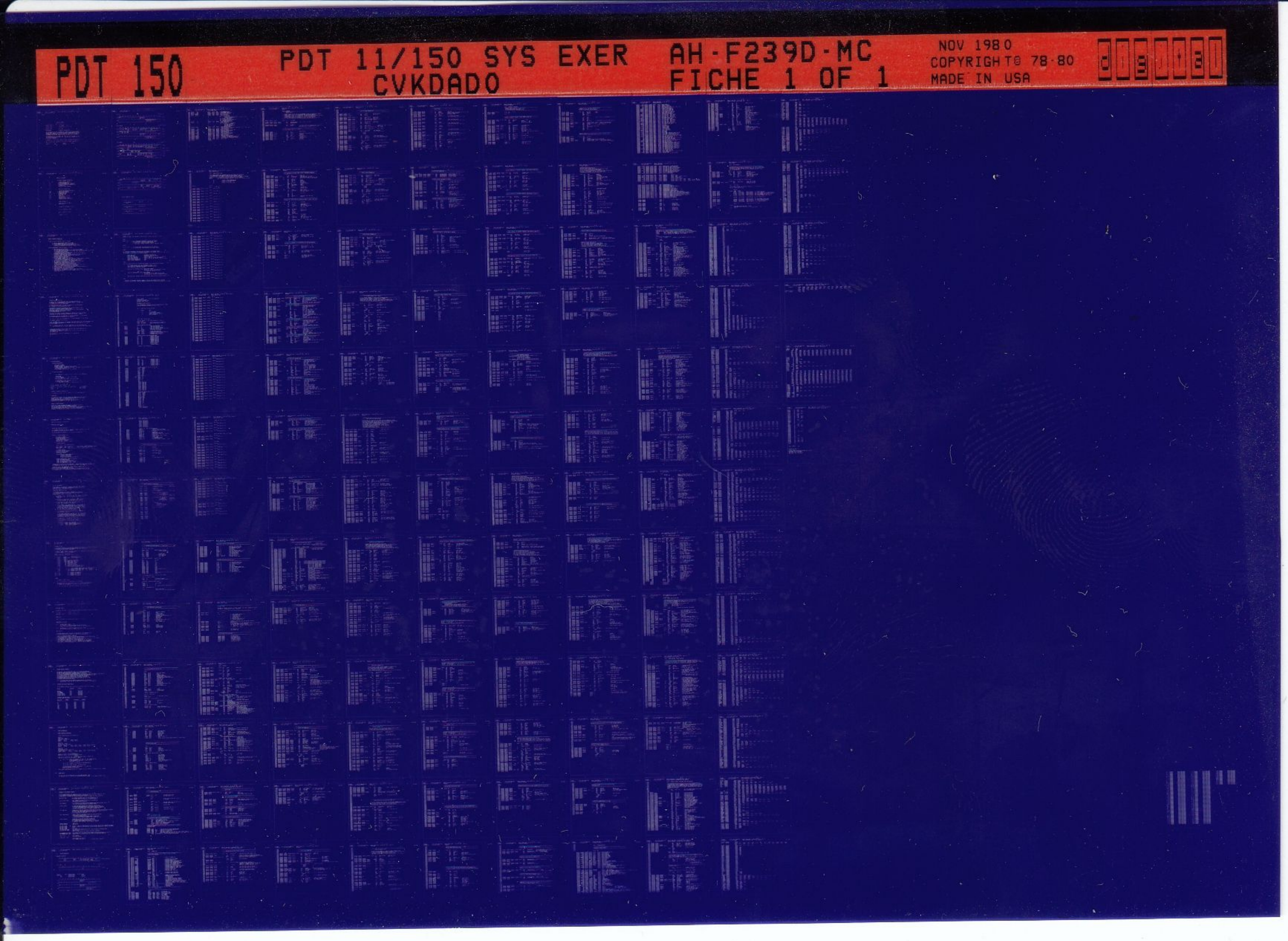
PDT 150

PDT 11/150 SYS EXER
CVKDADO

AH-F239D-MC
FICHE 1 OF 1



.REM %

IDENTIFICATION

PRODUCT CODE: AC-F238D-MC

PRODUCT NAME: CVKDAD0 PDT11/150 SYS EXER

PRODUCT DATE: JULY 1980

MAINTAINER: DIAGNOSTIC ENGINEERING

TABLE OF CONTENTS

1.0 GENERAL PROGRAM INFORMATION.

1.1 PROGRAM PURPOSE (ABSTRACT).
1.2 SYSTEM REQUIREMENTS.
1.3 RELATED DOCUMENTS AND STANDARDS.
1.4 DIAGNOSTIC HIERARCHY PREREQUISITES.
1.5 ASSUMMPTIONS.
1.6 RUNNING SYSTEMS PROGRAMS.

2.0 OPERATING INSTRUCTIONS.

2.1 LOADING AND STARTING PROCEDURES.
2.2 SPECIAL ENVIRONMENTS.
2.3 OPERATIONAL SWITCH SETTINGS
2.4 PROGRAM OPTIONS.
2.5 EXECUTION TIMES.
2.6 POWER FAIL.
2.7 COMPATABILITY TESTING.
2.8 COPY UTILITY.
2.9 RUNNING WITHOUT CONSOLE.
2.10 CHANGING DEFAULT BAUD RATES.

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.
3.2 ERROR HALTS.

4.0 PROGRESS & PERFORMANCE REPORTS.

5.0 DEVICE INFORMATION TABLES.

6.0 DEVICE CONNECTOR LAYOUT.

7.0 SUMMARY OF TESTS.

1.0 GENERAL PROGRAM INFORMATION.

PROGRAM BUG FIXES FROM REV CVKDAC TO CVKDAD:

A. REPLACE NON-EXISTANT ERROR 140 TO 130.
B. DO NOT EXAMINE KBD WHILE IN SYNC MODEM TEST.
C. DO ONLY WORST CASE PATTERNS ON COMPATABILITY TESTS
   OTHERWISE DO RANDOM/WCP PATTERNS ON ALTERNATE PASSES.

PROGRAM ENHANCEMENTS FROM REV CVKDAC TO CVKDAD:

A. NEW DISKETTE DATA PATTERNS.
   THE 1'ST WORD OF EACH SECTOR IS ITS TRACK/SECTOR NUMBER.
   THE REMAINING WORDS ARE FROM A PATCH-ABLE TABLE CONTAINING WORST CASE PATTERN
   ...OR RANDOM DATA PATTERNS.
   THE PATTERN ALTERNATES BETWEEN WORST CASE PATT ON ODD PASSES
   & RANDOM PATTERNS ON EVEN PASSES.
B. THE DEFAULT BAUD RATES ARE PRINTED ON THE 1'ST PASS STARTUP
   FOR THE CLUSTERS, PRINTER, SYNC & ASYNC COMM PORTS.
C. EXPANDED DISKETTE ERROR MESSAGES.
D. SET SYNC MODEM TO 9600 BAUD.
E. FLOPPY RESTORE COMMAND TEST (DXTST4):
   IF DX0 DROPPED, CHECK & TEST DX1 INSTEAD OF JUMPING OVER Dx1 PORTION.
F. CLEAR DEVCT AFTER INCREMENTING 'PASS' IN 'EOP' FOR APT.
G. SET MAX SEEKS TO 500 ONLY IF NOT DOING COMPAT TESTS.
H. ADD CONTROL-U FUNCTIONALITY TO INPUTTING A NEW DEVICE MAP.
I. INCREMENT DEVCT MORE FREQUENTLY FOR APT.
J. CHANGE SECTOR INTERLEAVING FROM EVERY OTHER SECTOR
   TO EVERY 3'RD SECTOR TO CATCH INTERLEAVING.
K. CLEAR ERROR COUNTERS ON RESTARTS OF COMPATABILITY TESTS 1 & 2.
L. ADDED FILL & EMPTY BUFFER TESTS TO DEVICE MAP.
M. IMPROVED ABILITY TO DISTINGUISH SEEK ERRORS FROM READ HEADER FAULTS.
N. TYPE PASS NUMBER WITH EACH ERROR REPORT.
O. ADDED ABILITY TO RUN WITHOUT CONSOLE.
P. ADDED ABILITY FOR OPERATOR TO EASILY MODIFY BAUD RATES.
Q. ADDED ABILITY TO INPUT UNIT SERIAL NUMBER & OUTPUT IT ON EACH ERROR REPORT
   & EACH END OF PASS.

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 LOCºBACK (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 (NOT REQ'D AFTER STARTUP)
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 CVKDAD.
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 & 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)

        CVKDAD  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)
        SERIAL NO. = 123        NEW = A123<CR>  (UP TO 7 CHARS ALLOWED)

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

        PROGRAM DEFAULTS (UNLESS MODIFIED):
                CLUSTERS @ 2400 BAUD
                PRINTER  @ 9600 BAUD
                SYNC & ASYNC COMM PORTS @ 9600 BAUD

        INSERT SCRATCH DISKS, TYPE 'P' FOR NORMAL TESTING
        '240G' FOR NORMAL RESTARTS
        '250G' TO COPY SYS EXERCISER DISK
        '260G' FOR COMPATABILITY PASS 1: WRITE
        '270G' FOR COMPATABILITY 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 PY 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
     5 SET =        40 = DROP FILL & EMPTY BUFFER TESTS
     4 SET =        20 = RUN WITHOUT A CONSOLE PRESENT

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 15-4 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, PERFORMANCE REPORTS NOT ENABLED:

1'ST PASS: APPROX. 1 MIN. 45 SEC.

SUBSEQUENT PASSES: APPROX. 10 MIN. 30 SEC.

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'.

2.9 RUNNING WITHOUT CONSOLE.

SETTING BIT 4 ON THE DEVICE MAP (SEC. 2.4) ALLOWS THE OPERATOR
TO RUN THE EXERCISER WITHOUT A CONSOLE ONCE THE PROGRAM IS UNDERWAY.
THE END OF PASS COUNTER WILL INCREMENT NORMALLY WHILE THE
END OF PASS & PERFORMANCE REPORT MESSAGES WILL BE SURPRESSED WHEN THE CONSOLE IS NOT PRESENT.
IF AN ERROR OCCURES, THE 2 USER LIGHTS WILL GO ON & THE PROCESSOR WILL HALT.
THIS CONDITION TELLS THE OPERATOR THAT AN ERROR HAS OCCURED.
THE OPERATOR THEN PLUGS A CONSOLE BACK IN & DOES A PROCEED,"P".
THE ERROR MESSAGE WILL PRINT OUT & TESTING RESUMES AT WHICH TIME
THE CONSOLE CAN BE UNPLUGGED AGAIN.

NOTE: THE CONSOLE MUST NOT BE REMOVED DURING PRINTOUT OF A MESSAGE.
DOING SO WILL CAUSE THE UNIT TO HANG UNTIL THE CONSOLE IS RE-CONNECTED.

2.10 CHANGING THE DEFAULT BAUD RATES.

THE FOLLOWING LOCATIONS HOLD THE DEFAULT BAUD RATES USED BY THE PROGRAM:

DEVICE	LOCATION	DEFAULT
PRINTER	3520	9600 BAUD
SYNC MODEM	3522	9600 BAUD
ASYNC MODEM	3524	9600 BAUD
CLUSTER TERM #1	3526	2400 BAUD
CLUSTER TERM #2	3530	2400 BAUD
CLUSTER TERM #3	3532	2400 BAUD

TO MODIFY THE BAUD RATES, LOAD THE FOLLOWING OCTAL NUMBERS
INTO THE ABOVE LOCATIONS:

BAUD RATE	NUMBER	BAUD RATE	NUMBER
50	0	1800	4000
75	400	2000	4400
110	1000	2400	5000
134.5	1400	3600	5400
150	2000	4800	6000
300	2400	7200	6400
600	3000	9600	7000
1200	3400	19200	7400

3.0 ERROR INFORMATION.

3.1 ERROR REPORTING PROCEDURE.

TYPICAL ERROR PRINTOUTS ARE SHOWN BELOW:

```
PASS # 2
SERIAL NO. = A123
TERM #3 DATA COMP ERR
ERROR #    ERR PC       EXPECT  RECVD
000011     006516       000200  000100

PASS # 5
SERIAL NO. = A123
DX1 SOFT ERR - DATA COMP
DXTST #    ERR PC       RXCS    RXES    RXSA    EXPECT  RECVD  # RETRIES
000003     010636       100337  000230  003405  100520  100120       4

PASS # 8
SERIAL NO. = A123
DX0 HARD ERR - AFTER WRITE CMD
ERROR # DXTST #     ERR PC       RXCS    RXES    TRACK   SECTOR  #RETRIES
000016  000001      011246       100337  000230    59      18       10
```

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

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

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

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

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

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

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

3.2 ERROR HALTS.

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

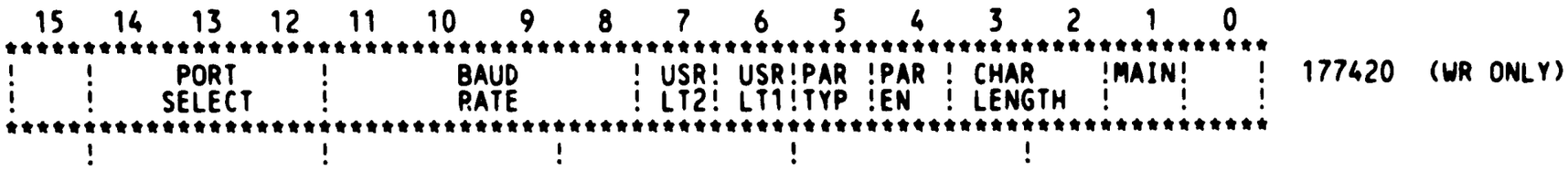
4.0 PROGRESS & PERFORMANCE REPORTS

THE FOLLOWING REPORTS ARE ENABLED 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 & CONSOLE CONNECTED):

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
SERIAL NO. = A123	TOTAL ERRORS THIS PASS: 0 SOFT ERRORS THIS PASS: 0 (...& ENTIRE PROCESS REPEATS)

5.0 DEVICE REGISTERS.

PARAMETER REGISTER:

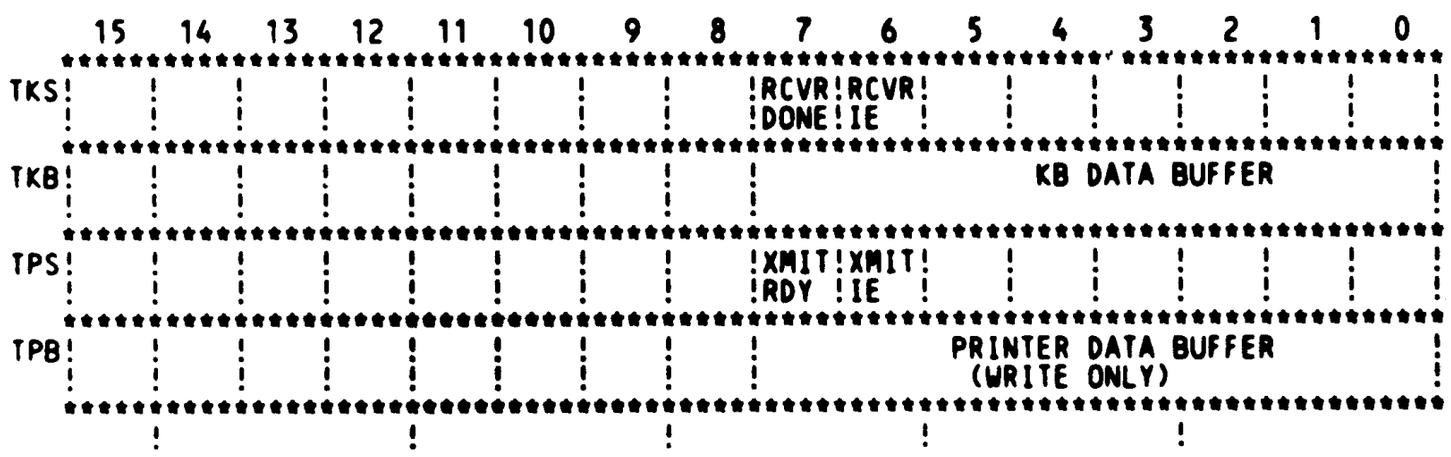


177420 (WR ONLY)

LINE CLOCK: VECTOR: 100

177546

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



PRINTER: VECTOR: 200

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

ASYNC MODEM: VECTOR: 330/334

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

SYNC MODEM: VECTOR: 340/344

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

DISK: VECTOR: 264

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

6.0 DEVICE CONNECTOR LAYOUT

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

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
TRACKS 0-4, 40-44, 72-76 ON THE 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 ERROR OR CRC ERROR.

%

```
                        .TITLE  CVKDAD          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
```

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

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

                                        ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                100000          BIT15=  100000
                040000          BIT14=  40000
                020000          BIT13=  20000
                010000          BIT12=  10000
                004000          BIT11=  4000
                002000          BIT10=  2000
                001000          BIT09=  1000
                000400          BIT08=  400
                000200          BIT07=  200
                000100          BIT06=  100
                000040          BIT05=  40
                000020          BIT04=  20
                000010          BIT03=  10
                000004          BIT02=  4
                000002          BIT01=  2
                000001          BIT00=  1
                                .EQUIV  BIT09,BIT9
                                .EQUIV  BIT08,BIT8
```

```
                        .EQUIV  BIT07,BIT7
                        .EQUIV  BIT06,BIT6
                        .EQUIV  BIT05,BIT5
                        .EQUIV  BIT04,BIT4
                        .EQUIV  BIT03,BIT3
                        .EQUIV  BIT02,BIT2
                        .EQUIV  BIT01,BIT1
                        .EQUIV  BIT00,BIT0

                        ;*BASIC "CPU" TRAP VECTOR ADDRESSES
000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
000014                  TBITVEC=14              ;;"T" BIT
000014                  TRTVEC= 14              ;;TRACE TRAP
000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
000024                  PWRVEC= 24              ;;POWER FAIL
000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
000034                  TRAPVEC=34              ;;"TRAP" TRAP
000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR


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

000200                  PRVEC=  200             ;PRINTER

000100                  CLKVEC= 100             ;LINE CLOCK

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

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

000264                  RXVEC=  264             ;DISK

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

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

177420                  PARAM=  177420          ;PARAMETER REGISTER             (WRITE ONLY)

177546                  CLK=    177546          ;LINE CLOCK

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

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

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

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

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

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

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

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

                        ;BIT 14-12  PORT SELECT
                        ;
010000                  CONSOL= 10000           ;SELECT CONSOLE TERMINAL
000000                  CT1=    0               ;       CLUSTER TERMINAL #1
050000                  CT2=    50000           ;                       #2
060000                  CT3=    60000           ;                       #3
020000                  PRT=    20000           ;       PRINTER PORT
030000                  AMOD=   30000           ;       ASYNC COMM PORT
030000                  SMOD=   30000           ;       SYNC 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, SYNC & 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
```

```
1027
1028                            ;;*****************************************************************
1029                            ;*      ASYNC MODEM BIT DEFINITIONS
1030                            ;;*****************************************************************
1031
1032                            ;RCSR
1033                            ;
1034            100000          DSI=    BIT15           ;DATA SET INTERRUPT
1035            040000          RING=   BIT14           ;RING
1036            020000          CTS=    BIT13           ;CLEAR TO SEND
1037            010000          CDET=   BIT12           ;CARRIER DETECTED
1038            002000          SREC=   BIT10           ;SECONDARY RECD/SUPERVISORY RECD
1039            001000          DSRDY=  BIT9            ;DATA SET RDY
1040            000200          DONE=   BIT7            ;RECVR DONE
1041            000100          IE=     BIT6            ;RECVR IE
1042            000040          DSIE=   BIT5            ;DATA SET IE
1043            000010          SXMIT=  BIT3            ;SECONDARY XMIT/SUPERV XMIT
1044            000004          RTS=    BIT2            ;REQ TO SEND
1045            000002          DTR=    BIT1            ;DATA TERMINAL RDY
1046
1047                            ;RBUF
1048                            ;
1049            100000          ERR=    BIT15           ;ERROR
1050            040000          ORERR=  BIT14           ;OVERRUN ERROR
1051            020000          FRERR=  BIT13           ;FRAMING ERROR
1052            010000          PARERR= BIT12           ;PARITY ERROR
1053
1054                            ;XCSR
1055                            ;
1056            000200          RDY=    BIT7            ;XMIT RDY
1057            000100          IE=     BIT6            ;XMIT IE
1058            000001          BREAK=  BIT0            ;BREAK
1059
```

```
                        ;;*****************************************************************
                        ;*      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)
```

```
1107
1108                            ;XCSR
1109                            ;
1110            100000          DNA=    BIT15           ;DATA NOT AVAIL
1111            010000          MM=     BIT12           ;MAINT MODE
1112            004000          MCLK=   BIT11           ;MAINT CLOCK
1113            000400          MR=     BIT8            ;MASTER RESET
1114            000200          RDY=    BIT7            ;XMIT RDY
1115            000100          IE=     BIT6            ;XMIT IE
1116            000040          DNAIE=  BIT5            ;DATA NOT AVAIL IE
1117            000020          SEND=   BIT4            ;SEND
1118            000010          HDUP=   BIT3            ;1=HALF DUPLEX, 0=FULL DUPLEX (DEFAULT = 0 FOR TESTS)
1119
1120
1121
1122
1123                            ;;**************************************************************
1124                            ;DISK BIT DEFINITIONS
1125                            ;;**************************************************************
1126
1127                            ;RXCS
1128                            ;
1129            100000          ERR=    BIT15           ;ERROR
1130            000200          DONE=   BIT7            ;CONTR RDY
1131            000100          IE=     BIT6            ;DRIVE IE
1132            000020          DX1=    BIT4            ;UNIT SELECT: 0=DX0, 1=DX1
1133            000020          USEL=   BIT4            ;UNIT SELECT
1134
1135                            ;FUNCTIONS (INCLUDES BIT0 = GO)
1136
1137            000001          FBUF=   1               ;FILL BUFFER
1138            000003          EBUF=   3               ;EMPTY BUFFER
1139            000005          WSEC=   5               ;WRITE SECTOR
1140            000007          RSEC=   7               ;READ SECTOR
1141            000011          INITAL= 11              ;INITIALIZE
1142            000013          RSTAT=  13              ;READ STATUS
1143            000015          WDDSEC= 15              ;WRITE DELETED DATA SECTOR
1144            000017          RESTOR= 17              ;RESTORE TO TRACK 0
1145
1146                            ;RXES: ERROR & STATUS
1147                            ;
1148            000200          RDY=    BIT7            ;DRIVE RDY
1149            000040          DD=     BIT5            ;DELETED DATA DET.
1150            000020          RNF=    BIT4            ;RECORD NOT FOUND
1151            000010          CRC=    BIT3            ;CRC ERROR
1152            000004          LDATA=  BIT2            ;LOST DATA
1153            000002          INVADR= BIT1            ;INVALID ADDR
1154            000001          ID=     BIT0            ;INITIALIZE DONE
1155
1156
```

```
1157
1158                                    ;;*****************************************************************
1159                                    ;*      SWITCH REGISTER
1160                                    ;;*****************************************************************
1161
1162            000174                          .=174
1163   000174   000000                 DISPREG:        .WORD   0       ;SOFTWARE DISPLAY REG
1164   000176   000000                 SWREG:  .WORD 0                 ;SOFTWARE SWITCH REG
1165
1166            000100                          .=100
1167   000100   000102                         102                     ;SETUP CLOCK VECTOR AREA TO DO RTI
1168   000102   000002                         2                       ;IF ENABLED
1169
1170            000200                          .=200
1171   000200   000137  003574                 JMP     START           ;USE ONLY ONCE. WILL BE OVERLAID BY
1172                                                                    ;PRINTER VECTOR ADDR.
1173            000240                          .=240
1174   000240   000137  003574                 JMP     START           ;RESTART ADDR
1175
1176            000250                          .=250
1177   000250   000137  003556                 JMP     START3          ;ENTER HERE FOR COPY UTILITY
1178            000260                          .=260
1179   000260   000137  003536                 JMP     START1          ;COMPATABILTIY PASS 1
1180            000270                          .=270
1181   000270   000137  003544                 JMP     START2          ;               PASS 2
1182
1183
1184
1185            001000                  .=1000
1186                                    .SBTTL  APT PARAMETER BLOCK
1187
1188                                    ;;*****************************************************************
1189                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1190                                    ;;*****************************************************************
1191            001000                          .$X=.   ;;SAVE CURRENT LOCATION
1192            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1193   000024   000200                         200     ;;FOR APT START UP
1194            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1195   000044   001000                         $APTHDR ;;POINT TO APT HEADER BLOCK
1196            001000                          .=.$X   ;;RESET LOCATION COUNTER
1197                                    ;;*****************************************************************
1198                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1199                                    ;INTERFACE SPEC.
1200
1201   001000                           $APTHD:
1202   001000   000000                 $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1203   001002   001170                 $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1204   001004   000024                 $TSTM:  .WORD   20.     ;;RUN TIM OF LONGEST TEST
1205   001006   000226                 $PASTM: .WORD   150.    ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1206   001010   000000                 $UNITM: .WORD   0       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1207   001012   000030                         .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```
                                .SBTTL  COMMON TAGS

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

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

                                ;;*********************************************************************
                                .EVEN
001170                  $MAIL:                          ;;APT MAILBOX
001170  000000          $MSGTY: .WORD   AMSGTY          ;;MESSAGE TYPE CODE
001172  000000          $FATAL: .WORD   AFATAL          ;;FATAL ERROR NUMBER
001174  000000          $TESTN: .WORD   ATESTN          ;;TEST NUMBER
001176  000000          $PASS:  .WORD   APASS           ;;PASS COUNT
001200  000000          $DEVCT: .WORD   ADEVCT          ;;DEVICE COUNT
001202  000000          $UNIT:  .WORD   AUNIT           ;;I/O UNIT NUMBER
001204  000000          $MSGAD: .WORD   AMSGAD          ;;MESSAGE ADDRESS
001206  000000          $MSGLG: .WORD   AMSGLG          ;;MESSAGE LENGTH
001210                  $ETABLE:                        ;;APT ENVIRONMENT TABLE
```

```
1264  001210  000              $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
1265  001211  000              $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
1266  001212  000000           $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
1267  001214  000000           $USWR:  .WORD   AUSWR   ;;USER SWITCHES
1268  001216  000000           $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
1269                            ;*                      BITS 15-11=CPU TYPE
1270                            ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
1271                            ;*                              11/70=06,PDQ=07,Q=10
1272                            ;*                      BIT 10=REAL TIME CLOCK
1273                            ;*                      BIT  9=FLOATING POINT PROCESSOR
1274                            ;*                      BIT  8=MEMORY MANAGEMENT
1275  001220  000              $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
1276  001221  000              $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
1277                            ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
1278                            ;*                              900 NSEC CORE=001
1279                            ;*                              300 NSEC BIPOLAR=002
1280                            ;*                              500 NSEC MOS=003
1281  001222  000000           $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
1282                            ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
1283  001224  000              $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
1284  001225  000              $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
1285  001226  000000           $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
1286  001230  000              $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
1287  001231  000              $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
1288  001232  000000           $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
1289  001234  000              $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
1290  001235  000              $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
1291  001236  000000           $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
1292  001240  000300           $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
1293  001242  000000           $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
1294  001244  176500           $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
1295  001246  000017           $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
1296  001250                   $ETEND:
1297                            .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  023324  026460  027070          EM1,DH2,DT2,DF
001256  027364
                                ;ERR 2
001260  023324  026460  027070          EM1,DH2,DT2,DF
001266  027364
                                ;ERR 3
001270  023343  026504  027100          EM2,DH3,DT3,DF
001276  027364
                                ;ERR 4
001300  023324  026460  027070          EM1,DH2,DT2,DF
001306  027364
                                ;ERR 5
001310  023343  026504  027100          EM2,DH3,DT3,DF
001316  027364
                                ;ERR 6
001320  023365  026441  027062          EM3,DH1,DT1,DF
001326  027364
                                ;ERR 7
001330  023431  026546  027114          EM4,DH4,DT4,DF
001336  027364
                                ;ERR 10
001340  023454  026577  027124          EM5,DH5,DT5,DF
001346  027364
                                ;ERR 11
001350  023502  026577  027136          EM6,DH5,DT6,DF
001356  027364
                                ;ERR 12
001360  023530  026577  027150          EM7,DH5,DT7,DF
001366  027364
                                ;ERR 13
001370  023556  026577  027162          EM8,DH5,DT8,DF
001376  027364
                                ;ERR 14
001400  023607  026577  027162          EM9,DH5,DT8,DF
001406  027364
                                ;ERR 15
001410  023637  026634  027174          EM10,DH6,DT9,DF
001416  027364
```

```
1353                                        ;ERR 16
1354  001420  023664  026701  027212              EM11,DH7,DT10,DF1
1355  001426  027372
1356                                        ;ERR 17
1357  001430  023716  026701  027212              EM12,DH7,DT10,DF1
1358  001436  027372
1359                                        ;ERR 20
1360  001440  023750  026701  027212              EM13,DH7,DT10,DF1
1361  001446  027372
1362                                        ;ERR 21
1363  001450  024001  026701  027212              EM14,DH7,DT10,DF1
1364  001456  027372
1365                                        ;ERR 22
1366  001460  024032  026770  027234              EM15,DH8,DT11,DF2
1367  001466  027402
1368                                        ;ERR 23
1369  001470  024066  026770  027234              EM16,DH8,DT11,DF2
1370  001476  027402
1371                                        ;ERR 24
1372  001500  024112  026701  027256              EM17,DH7,DT12,DF1
1373  001506  027372
1374                                        ;ERR 25
1375  001510  024144  026701  027256              EM18,DH7,DT12,DF1
1376  001516  027372
1377                                        ;ERR 26
1378  001520  024176  026701  027256              EM19,DH7,DT12,DF1
1379  001526  027372
1380                                        ;ERR 27
1381  001530  024227  026701  027256              EM20,DH7,DT12,DF1
1382  001536  027372
1383                                        ;ERR 30
1384  001540  024260  026770  027300              EM21,DH8,DT13,DF2
1385  001546  027402
1386                                        ;ERR 31
1387  001550  024314  026770  027300              EM22,DH8,DT13,DF2
1388  001556  027402
1389                                        ;ERR 32
1390  001560  024340  026441  027062              EM23,DH1,DT1,DF
1391  001566  027364
1392                                        ;ERR 33
1393  001570  024351  026546  027114              EM24,DH4,DT4,DF
1394  001576  027364
1395                                        ;ERR 34
1396  001600  024366  026441  027062              EM25,DH1,DT1,DF
1397  001606  027364
1398                                        ;ERR 35
1399  001610  024403  026441  027062              EM26,DH1,DT1,DF
1400  001616  027364
1401                                        ;ERR 36
1402  001620  024420  026441  027062              EM27,DH1,DT1,DF
1403  001626  027364
```

```
1404                                        ;ERR 37
1405  001630  024435  026441  027062              EM28,DH1,DT1,DF
1406  001636  027364
1407                                        ;ERR 40
1408  001640  024454  026441  027062              EM29,DH1,DT1,DF
1409  001646  027364
1410                                        ;ERR 41
1411  001650  024474  026634  027322              EM30,DH6,DT14,DF
1412  001656  027364
1413                                        ;ERR 42
1414  001660  024505  026634  027322              EM31,DH6,DT14,DF
1415  001666  027364
1416                                        ;ERR 43
1417  001670  024516  026441  027062              EM32,DH1,DT1,DF
1418  001676  027364
1419                                        ;ERR 44
1420  001700  024561  026441  027062              EM33,DH1,DT1,DF
1421  001706  027364
1422                                        ;ERR 45
1423  001710  024622  026441  027062              EM34,DH1,DT1,DF
1424  001716  027364
1425                                        ;ERR 46
1426  001720  024474  026634  027322              EM30,DH6,DT14,DF
1427  001726  027364
1428                                        ;ERR 47
1429  001730  024474  026634  027322              EM30,DH6,DT14,[
1430  001736  027364
1431                                        ;ERR 50
1432  001740  024474  026634  027322              EM30,DH6,DT14,DF
1433  001746  027364
1434                                        ;ERR 51
1435  001750  024474  026634  027322              EM30,DH6,DT14,DF
1436  001756  027364
1437                                        ;ERR 52
1438  001760  024661  026634  027174              EM35,DH6,DT9,DF
1439  001766  027364
1440                                        ;ERR 53
1441  001770  024726  026770  027234              EM36,DH8,DT11,DF2
1442  001776  027402
1443                                        ;ERR 54
1444  002000  024775  026634  027174              EM37,DH6,DT9,DF
1445  002006  027364
1446                                        ;ERR 55
1447  002010  025021  026634  027174              EM38,DH6,DT9,DF
1448  002016  027364
1449                                        ;ERR 56
1450  002020  024474  026634  027322              EM30,DH6,DT14,DF
1451  002026  027364
1452                                        ;ERR 57
1453  002030  025067  026634  027174              EM39,DH6,DT9,DF
1454  002036  027364
```

```
1455                                        ;ERR 60
1456  002040  024474  026634  027322              EM3C,DH6,DT14,DF
1457  002046  027364
1458                                        ;ERR 61
1459  002050  025117  026634  027174              EM40,DH6,DT9,DF
1460  002056  027364
1461                                        ;ERR 62
1462  002060  025162  026634  027174              EM41,DH6,DT9,DF
1463  002066  027364
1464                                        ;ERR 63
1465  002070  025216  026634  027174              EM42,DH6,DT9,DF
1466  002076  027364
1467                                        ;ERR 64
1468  002100  025237  026701  027212              EM43,DH7,DT10,DF1
1469  002106  027372
1470                                        ;ERR 65
1471  002110  025301  026701  027212              EM44,DH7,DT10,DF1
1472  002116  027372
1473                                        ;ERR 66
1474  002120  025341  026701  027212              EM45,DH7,DT10,DF1
1475  002126  027372
1476                                        ;ERR 67
1477  002130  025402  026701  027212              EM46,DH7,DT10,DF1
1478  002136  027372
1479                                        ;ERR 70
1480  002140  025441  026701  027256              EM47,DH7,DT12,DF1
1481  002146  027372
1482                                        ;ERR 71
1483  002150  025503  026701  027256              EM48,DH7,DT12,DF1
1484  002156  027372
1485                                        ;ERR 72
1486  002160  025543  026701  027256              EM49,DH7,DT12,DF1
1487  002166  027372
1488                                        ;ERR 73
1489  002170  025604  026701  027256              EM50,DH7,DT12,DF1
1490  002176  027372
1491                                        ;ERR 74
1492  002200  025643  026634  027174              EM51,DH6,DT9,DF
1493  002206  027364
1494                                        ;ERR 75
1495  002210  025667  026634  027322              EM52,DH6,DT14,DF
1496  002216  027364
1497                                        ;ERR 76
1498  002220  025713  026577  027340              EM53,DH5,DT15,DF
1499  002226  027364
1500                                        ;ERR 77
1501  002230  025713  026577  027352              EM53,DH5,DT16,DF
1502  002236  027364
```

```
1503
1504                                    ; ERRORS FOR COPY UTILITY
1505
1506                                    ;ERR 100
1507  002240  023750  026634  027322            EM13,DH6,DT14,DF
1508  002246  027364
1509                                    ;ERR 101
1510  002250  024112  026634  027322            EM17,DH6,DT14,DF
1511  002256  027364
1512                                    ;ERR 102
1513  002260  025762  026634  027322            EM54,DH6,DT14,DF
1514  002266  027364
1515                                    ;ERR 103
1516  002270  024176  026634  027322            EM19,DH6,DT14,DF
1517  002276  027364
1518
1519                                    ,BAD SECTOR/TRACK TABLE FULL ERRORS
1520
1521                                    ;ERR 104
1522  002300  025777  000000  000000            EM55,0,0,0
1523  002306  000000
1524                                    ;ERR 105
1525  002310  026034  000000  000000            EM56,0,0,0
1526  002316  000000
1527
1528                                    ;CRC ERRORS
1529
1530                                    ;ERR 106
1531  002320  026071  026701  027212            EM57,DH7,DT10,DF1
1532  002326  027372
1533                                    ;ERR 107
1534  002330  026126  026770  027234            EM58,DH8,DT11,DF2
1535  002336  027402
1536                                    ;ERR 110
1537  002340  026167  000000  000000            EM59,0,0,0
1538  002346  000000
1539                                    ;ERR 111
1540  002350  026225  026701  027256            EM60,DH7,DT12,DF1
1541  002356  027372
1542                                    ;ERR 112
1543  002360  026262  026770  027300            EM61,DH8,DT13,DF2
1544  002366  027402
1545                                    ;ERR 113
1546  002370  026323  000000  000000            EM62,0,0,0
1547  002376  000000
1548                                    ;ERR 114
1549  002400  025667  026634  027322            EM52,DH6,DT14,DF
1550  002406  027364
1551                                    ;ERR 115
1552  002410  025667  026634  027322            EM52,DH6,DT14,DF
1553  002416  027364
1554                                    ;ERR 116
1555  002420  025667  026634  027322            EM52,DH6,DT14,DF
1556  002426  027364
1557                                    ;ERR 117
1558  002430  024474  026634  027322            EM30,DH6,DT14,DF
```

```
                                                                    H 3
 1559  002436  027364
 1560                                  ;ERR 120
 1561  002440  024505  026634  027322          EM31,DH6,DT14,DF
 1562  002446  027364
 1563                                  ;ERR 121
 1564  002450  024474  026634  027322          EM30,DH6,DT14,DF
 1565  002456  027364
 1566                                  ;ERR 122
 1567  002460  024505  026634  027322          EM31,DH6,DT14,DF
 1568  002466  027364
 1569                                  ;ERR 123
 1570  002470  024505  026634  027322          EM31,DH6,DT14,DF
 1571  002476  027364
 1572                                  ;ERR 124
 1573  002500  025667  026634  027322          EM52,DH6,DT14,DF
 1574  002506  027364
 1575                                  ;ERR 125
 1576  002510  025667  026634  027322          EM52,DH6,DT14,DF
 1577  002516  027364
 1578
 1579
 1580                                  ;EIS FIS ERRORS
 1581
 1582
 1583                                  ;ERR 126
 1584  002520  026361  026441  027062          EM63,DH1,DT1,DF
 1585  002526  027364
 1586                                  ;ERR 127
 1587  002530  026416  026441  027062          EM64,DH1,DT1,DF
 1588  002536  027364
 1589
 1590                                  ;NORMAL ERRORS
 1591
 1592                                  ;ERR 130
 1593  002540  025216  026634  027174          EM42,DH6,DT9,DF
 1594  002546  027364
 1595                                  ;ERR 131
 1596  002550  025667  026634  027322          EM52,DH6,DT14,DF
 1597  002556  027364
 1598
```

```
1599                                    ;;*****************************************************************
1600                                    ;REGISTERS & BUFFERS FOR DX0 TESTS
1601                                    ;;*****************************************************************
1602
1603  002560  000000                    D0PAT:  0               ;DX0 PATT TEST DONE FLAG
1604  002562  000000                    D0CNT:  0               ;DX0 RANDOM SEEK COUNTER
1605  002564  000000                    D0ERR:  0               ;DX0 RCSR ERR COUNTER
1606  002566  000000                    D0TRK:  0               ;DX0 TRACK
1607  002570  000000                    D0SEC:  0               ;DX0 SECTOR
1608  002572  000000                    D0CMER: 0               ;DX0 DATA COMPARE ERROR FLAG
1609  002574  000000                    D0CERR: 0               ;DX0 TOTAL DATA COMP ERROR CT
1610  002576  000000                    D0EXP:  0               ;DX0 EXPECTED DATA
1611  002600  000000                    D0REC:  0               ;DX0 RECEIVED DATA
1612  002602  000000                    D0CRC:  0               ;DX0 CRC ERROR FLAG
1613  002604  000100                    D0BUF:  .BLKW 100       ;DX0 DATA BUFFER
1614  003004  000012                    D0BAD:  .BLKW 10.       ;DX0 BAD TRK/SEC TABLE
1615
1616
1617                                    ;;*****************************************************************
1618                                    ;REGISTERS & BUFFERS FOR DX1 TESTS
1619                                    ;;*****************************************************************
1620
1621  003030  000000                    D1PAT:  0               ;DX1 PATT TEST DONE FLAG
1622  003032  000000                    D1CNT:  0               ;DX1 RANDOM SEEK COUNTER
1623  003034  000000                    D1ERR:  0               ;DX1 RCSR ERR COUNTER
1624  003036  000000                    D1TRK:  0               ;DX1 TRACK
1625  003040  000000                    D1SEC:  0               ;DX1 SECTOR
1626  003042  000000                    D1CMER: 0               ;DX1 DATA COMPARE ERROR FLAG
1627  003044  000000                    D1CERR: 0               ;DX1 TOTAL DATA COMPARE ERROR CT
1628  003046  000000                    D1EXP:  0               ;DX1 EXPECTED DATA
1629  003050  000000                    D1REC:  0               ;DX1 RECEIVED DATA
1630  003052  000000                    D1CRC:  0               ;DX1 CRC ERROR FLAG
1631  003054  000100                    D1BUF:  .BLKW 100       ;DX1 DATA BUFFER
1632  003254  000012                    D1BAD:  .BLKW 10.       ;DX1 BAD TRK/SEC TABLE
```

```
1633
1634                                    ;;*****************************************************************
1635                                    ;PATTERN TABLE
1636                                    ;
1637                                    ;PATTERN IS "125252" FOR THE 1'ST PASS, RANDOM  FOR 2'ND & SUBSEQUENT PASSES.
1638                                    ;PATTERN IS  DETERMINED BY "SETPAT" ROUTINE.
1639                                    ;
1640                                    ;;*****************************************************************
1641
1642  003300  000077                    WCP:    .BLKW 77
1643
1644  003476  000000                    DXTST:  0               ;0 = FILL & EMPTY BUFFER TEST
1645                                                            ;1 = DATA PATT TEST
1646                                                            ;2 = RANDOM SEEK TEST
1647                                                            ;3 = INITIALIZE TEST
1648                                                            ;4 = RESTORE TEST
1649                                                            ;5 = WRITE DELETED DATA TEST
1650                                                            ;6 = INVALID ADDRESS TEST
1651
1652  003500  000000                    FIN:    0               ;COMMON FLG FOR DX0 & DX1 FOR TRK/SEC FINISHED
1653  003502  000000                    MAXTRK: 0               ;MAX TRACK #
1654                                                            ;0 FOR 1'ST PASS, 114 FOR SUBSEQUENT PASSES. SET AT START.
1655  003504  000000                    MAXSK:  0               ;# OF RANDOM SEEKS TO BE PERFORMED.
1656                                                            ;10 FOR 1'ST PASS, 500 FOR SUBSEQUENT PASSES. SET AT EOP.
1657  003506  000000                    COPFLG: 0               ; 0 = NORM TEST   1 = COPY UTILITY ACTIVE
1658  003510  000000                    COMP1:  0               ;1 = COMPAT TESTS, PASS 1
1659  003512  000000                    COMP2:  0               ;1 =               PASS 2
1660
1661  003514  000000                    SMFLG:  0               ;SYNC MODEM FLAG
1662  003516  000000                    SEKFLG: 0               ;SEEK ERROR FLAG
1663
1664
1665                                    ;;*****************************************************************
1666                                    ;DEFAULT BAUD RATE REGISTERS
1667                                    ;;*****************************************************************
1668
1669  003520  007000                    PRBAUD: B9600           ;PRINTER DEFAULT
1670  003522  007000                    SMBAUD: B9600           ;SYNC MODEM DEFAULT
1671  003524  007000                    AMBAUD: B9600           ;ASYNC MODEM DEFAULT
1672  003526  005000                    C1BAUD: B2400           ;CLUSTER # 1 DEFAULT
1673  003530  005000                    C2BAUD: B2400           ;CLUSTER # 2 DEFAULT
1674  003532  005000                    C3BAUD: B2400           ;CLUSTER # 3 DEFAULT
1675  003534  000000                    HOLD:   0               ;TEMP HOLDING REG
1676
1677
```

```
                                        .SBTTL  PROGRAM

003536  005237  003510         START1: INC     COMP1           ;COMPAT PASS 1
003542  000424                         BR      S1

003544  005237  003512         START2: INC     COMP2           ;COMPAT PASS 2
003550  005037  003510                 CLR     COMP1
003554  000421                         BR      S2

003556  005237  003506         START3: INC     COPFLG          ;COPY UTILITY
003562  005037  003510                 CLR     COMP1
003566  005037  003512                 CLR     COMP2
003572  000414                         BR      S3

                                        ;NORMAL START AND RESTART
003574  012737  000102  000100 START:  MOV     #CLKVEC+2,CLKVEC        ;DO RTI IF CLOCK INTEPR
003602  012737  000002  000102         MOV     #RTI,CLKVEC+2

003610  005037  003110                 CLR     COMP1           ;NORMAL TESTING
003614  005037  003112         S1:     CLR     COMP2
003620  005037  003506         S2:     CLR     COPFLG

003624  012706  001100         S3:     MOV     #STACK,SP       ;SETUP STACK POINTER
003630  042777  000100  175306         BIC     #100,@$TKS      ;DISABLE ANY TTY INTERRUPTS
                                                                ;TO RT-11 MONITOR
                                        ;MTPS   #PR4            ;DISABLE INTERRUPTS
003636  106427                         .WORD   106427
003640  000200                         PR4
003642  013737  000042  005054         MOV     42,HLD42        ;SAVE
003650  123727  001210  000001         CMPB    $ENV,#1         ;APT?
003656  001402                         BEQ     2$              ;BR IF YES
003660  005037  000042                 CLR     42              ;ELSE CLR SO WE CAN USE SOFTSWR
003664                          2$:
                                        .SBTTL  INITIALIZE THE COMMON TAGS
                                        ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
003664  012706  001100                 MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
003670  005026                         CLR     -(R6)+          ;;CLEAR MEMORY LOCATION
003672  022706  001140                 CMP     #SWR,R6 ;;DONE?
003676  001374                         BNE     .-6             ;;LOOP BACK IF NO
003700  012706  001100                 MOV     #STACK,SP       ;;SETUP THE STACK POINTER
                                        ;;INITIALIZE A FEW VECTORS
003704  012737  031004  000030         MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
003712  012737  000340  000032         MOV     #340,@#EMTVEC+2 ;;LEVEL 7
003720  012737  032114  000034         MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
003726  012737  000340  000036         MOV     #340,@#TRAPVEC+2;LEVEL 7
                                        ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
                                        ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
003734  013746  000004                 MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
003740  012737  003774  000004         MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
003746  012737  177570  001140         MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
003754  012737  177570  001142         MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
003762  022777  177777  175150         CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
003770  001012                         BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
                                                                ;;AND  THE HARDWARE SWR IS NOT = -1
003772  000403                         BR      65$             ;;BRANCH IF NO TIMEOUT
003774  012716  004002         64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
```

```
1734  004000  000002                          RTI
1735  004002  012737  000176  001140  65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
1736  004010  012737  000174  001142          MOV     #DISPREG,DISPLAY
1737  004016  012637  000004          66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
1738
1739  004022  005037  001176                  CLR     $PASS           ;;CLEAR PASS COUNT
1740  004026  132737  000200  001211          BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
1741  004034  001403                          BEQ     67$             ;;YES,USE NON-APT SWITCH
1742  004036  012737  001212  001140          MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
1743  004044                          67$:
1744                                  .SBTTL  TYPE PROGRAM NAME
1745                                  ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
1746  004044  005227  177777                  INC     #-1             ;;FIRST TIME?
1747  004050  001046                          BNE     68$             ;;BRANCH IF NO
1748  004052  104401  004120                  TYPE    ,69$            ;;TYPE ASCIZ STRING
1749                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
1750  004056  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
1751  004062  001012                          BNE     70$             ;;BRANCH IF YES
1752  004064  123727  001210  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
1753  004072  001406                          BEQ     70$             ;;BRANCH IF YES
1754  004074  023727  001140  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
1755  004102  001005                          BNE     71$             ;;BRANCH IF NO
1756  004104  104406                          GTSWR                   ;;GET SOFT-SWR SETTINGS
1757  004106  000403                          BR      71$
1758  004110  112737  000001  001134  70$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
1759  004116                          71$:
1760  004116  000423                          BR      68$             ;;GET OVER THE ASCIZ
1761                                  ;;69$:  .ASCIZ  <CRLF>*CVKDAD  PDT-11/150 SYSTEM EXERCISER*<CRLF>
1762  004166                          68$:
1763
1764  004166  005227  177777                  INC     #-1             ;1'ST TIME?
1765  004172  001401                          BEQ     3$              ;BR IF YES
1766  004174  104406                          GTSWR
1767
1768  004176  012737  000004  003502  3$:     MOV     #4,MAXTRK       ;TRK 0-4, 40-44, 72-76(10) FOR 1'ST PASS
1769  004204  012737  000012  003504          MOV     #10.,MAXSK      ;10 RANDOM SEEKS FOR 1'ST PASS
1770
1771  004212  005037  031432                  CLR     TERR            ;TOTAL ERR CT
1772  004216  005037  031434                  CLR     TSERR           ;TOTAL SOFT ERR CT FOR EOP TYPEOUT
1773
1774  004222  005737  003506                  TST     COPFLG          ;COPY UTILITY?
1775  004226  001056                          BNE     LOOP            ;BR IF YES
1776
```

```
1777                                    ;;*****************************************************************
1778                                    ;       DEVICE MAP
1779                                    ;;*****************************************************************
1780
1781  004230  123727  001210  000001          CMPB    $ENV,#1         ;APT?
1782  004236  001413                          BEQ     4$              ;BR IF YES
1783  004240  104412                          GTDEVM                  ;SHOW CURRENT DEVM & GET NEW
1784
1785                                    ;;*****************************************************************
1786                                    ;       GET SYSTEM SERIAL NUMBER
1787                                    ;;*****************************************************************
1788
1789  004242  105037  030473                  CLRB    $TTYIN+7        ;MAKE SURE NULL AT END OF INITIAL LOAD
1790  004246  104401  022614                  TYPE    ,SERIAL         ;SHOW CURRENT SERIAL #
1791  004252  104401  030464                  TYPE    ,$TTYIN
1792
1793  004256  104401  030524                  TYPE    ,$MNEW          ;GET NEW
1794  004262  104411                          RDLIN
1795  004264  005726                          TST     (SP)+           ;CLEAR STACK, SERIAL # ALWAYS IN TTYIN
1796
1797                                    ;;*****************************************************************
1798                                    ;       CHECK FOR COMPATABILITY TESTING
1799                                    ;;*****************************************************************
1800  004266  005737  003510          4$:     TST     COMP1           ;COMPAT PASS 1?
1801  004272  001034                          BNE     LOOP            ;BR IF YES
1802  004274  005737  003512                  TST     COMP2           ;COMPAT PASS 2?
1803  004300  001031                          BNE     LOOP            ;BR IF YES
1804
1805  004302  052737  000006  176610          BIS     #<RTS!DTR>,AMRC
1806  004310  042737  000006  176610          BIC     #<RTS!DTR>,AMRC ;DTR MUST BE TOGGLED.
1807
1808                                    ;;*****************************************************************
1809                                    ;       SIZE SYSTEM & SHOW DEFAULTS
1810                                    ;;*****************************************************************
1811
1812  004316  004737  012106                  JSR     PC,SIZE         ;SEE WHO IS ON THE SYSTEM
1813  004322  104401  021746                  TYPE    ,DEFBD          ;SHOW ALL DEFAULT BAUD RATES
1814  004326  123727  001210  000001          CMPB    $ENV,#1         ;/RE WE RUNNING UNDER APT?
1815  004334  001413                          BEQ     LOOP            ;BR IF YES & DONT HALT
1816  004336  032737  001000  001246          BIT     #1000,$DEVM     ;DX0 THERE?
1817  004344  001404                          BEQ     1$              ;BR IF YES
1818  004346  032737  000400  001246          BIT     #400,$DEVM      ;DX1 THERE?
1819  004354  001003                          BNE     LOOP            ;BR IF NO
1820  004356  104401  022150          1$:     TYPE    ,WARNING        ;INSERT SCRATCH DISKS
1821  004362  000000                          HALT
1822
```

```
1823                                    ;;*****************************************************************
1824                                    ;       NOW WE CAN DO SOME SERIOUS TESTING
1825                                    ;;***************************** *********************************
1826
1827  004364  012706  001100          LOOP:   MOV     #1100,SP        ;LOOP HERE AFTER EOP & RESET STACK
1828
1829  004370  004737  013620                  JCR     PC,TIMER1       ;TIMER TO ALLOW 'P' TO PRINT BEFORE RESET
1830  004374  000005                  RESET                           ;CLEAR ALL INTERRUPT ENABLES ONLY ON 1'ST PASS
1831  004376  042737  000100  177546  3$:     BIC     #IE,CLK         ;RESET SETS CLK IE. DONT ALLOW YET
1832
1833                                   ;MTPS   #PR0             ;ALLOW INTERRUPTS
1834  004404  106427                  .WORD   106427
1835  004406  000000                  PR0
1836
1837  004410  052737  000006  176610          BIS     #<RTS!DTR>,AMRC ;TO CONDITION ALL TERMINALS &
1838                                                                  ;COMM PORT TO OPERATE IN EXT LOOPBACK MODE.
1839                                                                  ;NO HARM IF LOOPBACK NOT USED.
1840  004416  004737  013620                  JSR     PC,TIMER1       ;TIMER TO ALLOW PREV XFERS TO FINISH
1841
1842  004422  005037  031436                  CLR     PERR            ;PASS ERR COUNT FOR EOP TYPEOUT
1843  004426  005037  031440                  CLR     PSERR           ;PASS SOFT ERR COUNT FOR EOP TYPEOUT
1844
1845  004432  004737  013716                  JSR     PC,CLRBAD       ;CLEAR BAD SECTOR/TRACK TABLES
1846  004436  004737  012644                  JSR     PC,SETPAT       ;LOAD 'WCP'' TABLE WITH PATTERN
1847
1848  004442  005737  003510                  TST     COMP1           ;COMPAT PASS 1?
1849  004446  001003                          BNE     1$              ;BR IF YES
1850  004450  005737  003512                  TST     COMP2           ;COMPAT PASS 2?
1851  004454  001405                          BEQ     2$              ;BR IF NO
1852
1853  004456  012737  000114  003502  1$:     MOV     #114,MAXTRK
1854  004464  000137  007750                  JMP     DXTST1          ;COMPATABILITY TESTING
1855
1856  004470  005737  003506          2$:     TST     COPFLG          ;COPY UTILITY?
1857  004474  001402                          BEQ     MEMTST          ;BR IF NO...NORMAL TESTING
1858  004476  000137  020714                  JMP     COPY            ;ELSE GO COPY
1859
```

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

004502  004737  013104          MEMTST: JSR     PC,EXAMKB

004506  012737  014174  000004          MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
004514  012737  000200  000006          MOV     #200,ERRVEC+2

004522  013703  012622                  MOV     MAXMEM,R3
004526  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'
                                ;;*****************************************************************

004532  005004                  1$:     CLR     R4              ;ADDR POINTER
004534  005037  014202                  CLR     INTFLG
004540  005714                          TST     (R4)            ;TEST LOCS 0 THRU END OF PGM.
004542  005737  014202                  TST     INTFLG          ;TIMEOUT?
004546  001403                          BEQ     2$              ;BR IF NO
004550  010437  005052                  MOV     R4,ADDR         ;FOR ERR TYP
004554  104001                          ERROR   1               ;TIMEOUT ON READ
004556  005724                  2$:     TST     (R4)+           ;BUMP ADDR
004560  020427  032276                  CMP     R4,#LSTAD       ;AT END?
004564  001363                          BNE     1$              ;BR IF NO
```

```
                        ;;*****************************************************************
                        ; TEST READ/WRITE PATTERNS FROM 'LSTAD' TO BOTTOM OF MONITOR
                        ;;*****************************************************************

004566  005005                  CLR     R5              ;PASS CTR
004570  012737  125252  005050  MOV     #125252,PAT     ;DATA PATT FOR 1'ST PASS
004576  012704  032276  8$:     MOV     #LSTAD,R4       ;ADDR POINTER. R/W LOCS LSTAD THRU
                                                        ;BOT OF RT11 OR BOT OF ABS LOADER
004602  005037  014202  3$:     CLR     INTFLG
004606  013714  005050          MOV     PAT,(R4)        ;WRITE
004612  005737  014202          TST     INTFLG          ;TIMEOUT?
004616  001403                  BEQ     4$              ;BR IF NO
004620  010437  005052          MOV     R4,ADDR         ;FOR ERR TYP
004624  104002                  ERROR   2               ;TIMEOUT ON WRITE
004626  011437  005046  4$:     MOV     (R4),MEMHLD     ;READ
004632  023737  005046  005050  CMP     MEMHLD,PAT      ;COMPARE
004640  001403                  BEQ     6$
004642  010437  005052          MOV     R4,ADDR         ;FOR ERR TYP
004646  104003                  ERROR   3               ;COMPARE ERROR

004650  005724          6$:     TST     (R4)+           ;BUMP POINTER
004652  004737  005014          JSR     PC,BOTMON       ;SEE IF AT BOTTOM OF MONITOR
004656  000751                  BR      3$              ;RET HERE IF NO

004660  005705                  TST     R5              ;ELSE HERE
004662  001005                  BNE     9$
004664  005205                  INC     R5
004666  012737  052525  005050  MOV     #052525,PAT     ;DATA PATT FOR 2'ND PASS
004674  000740                  BR      8$              ;REPEAT

                        ;;*****************************************************************
                        ;ADDRESS TEST. WRITE EACH LOC. WITH ITS ADDR. AS DATA. THEN VERIFY.
                        ;;*****************************************************************

004676  012704  032276  9$:     MOV     #LSTAD,R4       ;ADDR REG

004702  010414          10$:    MOV     R4,(R4)         ;LOAD ADDR AS DATA
004704  005724                  TST     (R4)+           ;BUMP ADDR
004706  004737  005014          JSR     PC,BOTMON       ;SEE IF AT BOT OF MON
004712  000773                  BR      10$             ;RET HERE IF NO

                                                        ;VERIFY PASS
004714  012704  032276          MOV     #LSTAD,R4
004720  020414          11$:    CMP     R4,(R4)         ;READ & CHECK
004722  001407                  BEQ     12$             ;BR IF OK
004724  010437  005052          MOV     R4,ADDR
004730  010437  005050          MOV     R4,PAT
004734  011437  005046          MOV     (R4),MEMHLD
004740  104005                  ERROR   5               ;MISCOMPARE

004742  005724          12$:    TST     (R4)+           ;BUMP ADDR
004744  004737  005014          JSR     PC,BOTMON       ;SEE IF AT BOT OF MON
004750  000763                  BR      11$             ;RET HERE IF NO

004752  012737  000006  000004  MOV     #ERRVEC+2,ERRVEC        ;RESTORE TIMEOUT VECTOR
004760  005037  000006          CLR     ERRVEC+2
```

```
1951
1952  004764  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
1953  004770  000432                             BR   EISFIS          ;RET HERE IF NOT
1954  004772  032777  010000  174140          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
1955  005000  001426                          BEQ     EISFIS
1956  005002  104401  022561                  TYPE    ,MEMORY
1957  005006  104401  022574                  TYPE    ,DUN
1958  005012  000421                          BR      EISFIS
1959
1960                                  ;;*****************************************************************
1961                                  ;ROUTINE TO DETERMINE IF AT BOTTOM OF RT-11 OR XXDP MONITOR
1962                                  ;;*****************************************************************
1963
1964  005014  023727  005054  001000  BOTMON: CMP     HLD42,#1000     ;RT-11 IF LOC 42 = 1000
1965  005022  001403                          BEQ     1$              ;BR IF RT-11
1966  005024  020403                          CMP     R4,R3           ;ELSE AT BOT OF XXDP MON?
1967  005026  001006                          BNE     3$              ;BR IF NO
1968  005030  000403                          BR      2$
1969
1970  005032  020437  000054          1$:     CMP     R4,54           ;LOC 54 CONTAINS LOW ADDR OF RT-11 MON.
1971  005036  001002                          BNE     3$              ;BR IF NOT THERE
1972  005040  062716  000002          2$:     ADD     #2,(SP)         ;RET+2 IF AT END
1973  005044  000207                  3$:     RTS     PC
1974
1975  005046  000000                  MEMHLD: 0                       ;PUT MEMORY READ HERE
1976  005050  000000                  PAT:    0                       ;PATT TO BE WRITTEN & READ
1977  005052  000000                  ADDR:   0                       ;ADDR UNDER TEST
1978  005054  000000                  HLD42:  0                       ;SAVE LOC 42
```

```
                                ;;*****************************************************************
                                ;                EIS/FIS TEST SECTION
                                ;;*****************************************************************
005056  012737  001750  005730  EISFIS: MOV     #1000.,EISCNT   ;LOAD EXECUTION LOOP COUNTER
005064  005737  012612                  TST     EIPRES          ;TEST IF SIZER DETERMINES EIS-FIS PRESENT
005070  001404                          BEQ     1$              ;BR IF NOT PRESENT
005072  032737  000100  001246          BIT     #BIT6,$DEVM     ;TEST IF EIS/FIS IS ENABLED
005100  001402                          BEQ     ASH1            ;BR IF YES
005102  000137  005760          1$:     JMP     NOEIS           ;BYPASS THIS SECTION - NOT SELECTED
                                ;;*****************************************************************
                                ;TEST ASH LEFT INSTRUCTION
                                ;;*****************************************************************
005106  012702  125252          ASH1:   MOV     #125252,R2      ;LOAD VALUE TO BE SHIFTED
005112  000257                          CCC                     ;CLEAR CONDITION CODES
005114  000264                          SEZ                     ;SET Z BIT
005116  000270                          SEN                     ;SET N BIT
005120  072227  000001                  ASH     #1,R2           ;SHIFT LEFT 1, RESULT = 52524
                                                                ;N=0, Z=0, V=1, C=1
005124  100403                          BMI     1$              ;IF N SET, REPORT ERROR
005126  001402                          BEQ     1$              ;IF Z SET, REPORT ERROR
005130  102001                          BVC     1$              ;IF V = 0, REPORT ERROR
005132  103401                          BCS     2$              ;IF C SET, GO ON WITH TEST
005134  104126          1$:     ERROR   126             ;INCORRECT CONDITION CODES FOR "ASH LEFT"
005136  022702  052524          2$:     CMP     #52524,R2       ;RESULT CORRECT ?
005142  001401                          BEQ     ASH2            ;BR IF OK
005144  104127                          ERROR   127             ;INCORRECT DATA FOR "ASH LEFT"
                                ;;*****************************************************************
                                ;TEST ASH RIGHT INSTRUCTION
                                ;;*****************************************************************
005146  012701  052525          ASH2:   MOV     #52525,R1       ;LOAD VALUE TO BE SHIFTED
005152  000257                          CCC                     ;CLEAR CONDITION CODES
005154  072127  177777                  ASH     #-1,R1          ;SHIFT RIGHT 1, RESULT = 25252
005160  022701  025252                  CMP     #25252,R1       ;RESULT CORRECT ?
005164  001401                          BEQ     ASH3            ;BR IF OK
005166  104127                          ERROR   127             ;INCORRECT DATA FOR "ASH RIGHT"
                                ;;*****************************************************************
                                ;TEST ASH LEFT SHIFT 000001 BY 16 SETS C.
                                ;;*****************************************************************
005170  012700  000001          ASH3:   MOV     #1,R0           ;LOAD VALUE TO BE SHIFTED
005174  000241                          CLC                     ;CLEAR C BIT
005176  072027  000020                  ASH     #16.,R0         ;SHIFT LEFT 16 TIMES
005202  103001                          BCC     1$              ;BR IF ERROR
005204  001401                          BEQ     ASH4            ;BR IF DATA CORRECT
005206  104126          1$:     ERROR   126             ;INCORRECT CONDITION CODES FOR "ASH LEFT"
                                ;;*****************************************************************
                                ;TEST ASH RIGHT SHIFT 100000 BY 16 SETS C.
                                ;;*****************************************************************
005210  012700  100000          ASH4:   MOV     #100000,R0      ;LOAD VALUE TO BE SHIFTED
005214  000241                          CLC                     ;CLEAR C BIT
005216  072027  177760                  ASH     #-16.,R0        ;SHIFT RIGHT 16 TIMES
005222  103401                          BCS     1$              ;BR IF NO ERROR
005224  104126                          ERROR   126             ;INCORRECT CONDITION CODE FOR "ASH RIGHT"
005226  020027  177777          1$:     CMP     R0,#177777      ;CHECK DATA
005232  001401                          BEQ     ASHC1           ;BR IF OK
005234  104127                          ERROR   127             ;INCORRECT DATA FOR "ASH RIGHT"
```

```
2034
2035                                    ;;*****************************************************************
2036                                    ;TEST ASHC LEFT INSTRUCTION
2037                                    ;;*****************************************************************
2038  005236  005002                    ASHC1:  CLR     R2              ;CLEAR MSW REGISTER
2039  005240  012703  125252                    MOV     #125252,R3      ;LOAD VALUE TO BE SHIFTED
2040  005244  000257                            CCC                     ;CLEAR CONDITION CODES
2041  005246  000264                            SEZ                     ;SET Z BIT
2042  005250  000261                            SEC                     ;SET C BIT
2043  005252  073227  000020                    ASHC    #16.,R2         ;SHIFT LEFT R3 INTO R2
2044                                                                    ;RESULT: N=1,Z=0,V=1,C=0
2045  005256  100003                            BPL     1$              ;IF N = 0, REPORT ERROR
2046  005260  001402                            BEQ     1$              ;IF Z SET, REPORT ERROR
2047  005262  102001                            BVC     1$              ;IF V = 0, REPORT ERROR
2048  005264  103001                            BCC     2$              ;IF C = 0, OK
2049  005266  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR ASHC LEFT
2050  005270  022702  125252            2$:     CMP     #125252,R2      ;R2 OK
2051  005274  001401                            BEQ     3$              ;BR IF CORRECT
2052  005276  104127                            ERROR   127             ;INCORRECT DATA FOR ''ASHC LEFT''
2053  005300  005703                    3$:     TST     R3              ;R3 OK
2054  005302  001401                            BEQ     ASHC2           ;BR IF CORRECT
2055  005304  104127                            ERROR   127             ;INCORRECT DATA FOR ''ASHC LEFT''
2056
2057                                    ;;*****************************************************************
2058                                    ;TEST ASHC RIGHT INSTRUCTION
2059                                    ;;*****************************************************************
2060  005306  005003                    ASHC2:  CLR     R3              ;CLEAR MSW REGISTER
2061  005310  012702  125252                    MOV     #125252,R2      ;LOAD VALUE TO BE SHIFTED
2062  005314  000257                            CCC                     ;CLEAR CONDITION CODES
2063  005316  000264                            SEZ                     ;SET Z BIT
2064  005320  000262                            SEV                     ;SET V BIT
2065  005322  000261                            SEC                     ;SET C BIT
2066  005324  073227  177760                    ASHC    #-16.,R2        ;SHIFT RIGHT 16 PLACES
2067                                                                    ;RESULT: N=1,Z=V=C=0
2068  005330  100002                            BPL     1$              ;IF N =0,REPORT ERROR
2069  005332  101401                            BLOS    1$              ;IF C OR Z SET, REPORT ERROR
2070  005334  102001                            BVC     2$              ;IF V = 0, GO ON WITH TEST
2071  005336  104126                    1$:     ERROR   126             ;INCORRECT CONDITION CODES FOR ''ASHC RIGHT''
2072  005340  022702  177777            2$:     CMP     #-1,R2          ;TEST R2
2073  005344  001401                            BEQ     3$              ;BR IF CORRECT
2074  005346  104127                            ERROR   127             ;INCORRECT DATA FOR ''ASHC RIGHT''
2075  005350  022703  125252            3$:     CMP     #125252,R3      ;TEST R2 TO R3 SHIFT
2076  005354  001401                            BEQ     ASHC3           ;BR IF CORRECT
2077  005356  104127                            ERROR   127             ;INCORRECT DATA FOR ''ASHC RIGHT''
```

```
                                        ;;*****************************************************************
                                        ;TEST ASHC LEFT SHIFT 00000,0002 BY 31 SETS C.
                                        ;;*****************************************************************
                                        ;
005360  005000          ASHC3:  CLR     R0              ;CLEAR MSW REGISTER
005362  012701  000002          MOV     #2,R1           ;LOAD VALUE TO BE SHIFTED
005366  073027  000037          ASHC    #31.,R0         ;LEFT SHIFT 0002 BY 31.
005372  103001                  BCC     1$              ;BR IF C BIT CLEARED
005374  001401                  BEQ     ASHC4           ;BR IF RESULT =0
005376  104126          1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR "ASHC LEFT"

                                        ;;*****************************************************************
                                        ;TEST ASHC RIGHT SHIFT 100000,00000 BY 32 SETS C.
                                        ;;*****************************************************************
                                        ;
005400  012700  100000  ASHC4:  MOV     #100000,R0      ;LOAD VALUE TO BE SHIFTED
005404  005001                  CLR     R1              ;CLEAR LSW REGISTER
005406  073027  000040          ASHC    #32.,R0         ;SHIFT RIGHT BY 32.
005412  103001                  BCC     1$              ;BR IF C BIT CLEARED
005414  100401                  BMI     2$              ;BR IF N BIT SET
005416  104126          1$:     ERROR   126             ;INCORRECT CONDITION CODE FOR "ASHC RIGHT"
005420  022701  177777  2$:     CMP     #-1,R1          ;CHECK LOW ORDER RESULT
005424  001003                  BNE     3$              ;BR IF IN ERROR
005426  022700  177777          CMP     #-1,R0          ;CHECK HIGH ORDER RESULT
005432  001401                  BEQ     MUL0            ;BR IF CORRECT
005434  104127          3$:     ERROR   127             ;INCORRECT DATA FOR "ASHC RIGHT"
```

```
2103                                    ;;*****************************************************************
2104                                    ;TEST MUL INSTRUCTION
2105                                    ;;*****************************************************************
2106                                    ;       125252*40000=165252,100000
2107  005436  012700  125252   MUL0:    MOV     #125252,R0      ;LOAD MULTIPLICAND
2108  005442  070027  040000            MUL     #40000,R0       ;R1:R0=40000*125252
2109  005446  100003                    BPL     1$              ;BR IF N BIT ERROR
2110  005450  102402                    BVS     1$              ;BR IF V BIT ERROR
2111  005452  001401                    BEQ     1$              ;BR IF Z BIT ERROR
2112  005454  103401                    BCS     2$              ;BR IF C BIT CORRECT
2113  005456  104126   1$:      ERROR   126             ;INCORRECT CONDITION CODE FOR 'MUL''
2114  005460  022700  165252   2$:      CMP     #165252,R0      ;HIGH ORDER RESULT OK ?
2115  005464  001003                    BNE     3$              ;BR IF NO
2116  005466  022701  100000            CMP     #100000,R1      ;LOW ORDER RESULT OK?
2117  005472  001401                    BEQ     DIV0            ;BR IF YES
2118  005474  104127   3$:      ERROR   127             ;INCORRECT DATA FOR 'MUL''
2119
2120
2121                                    ;;*****************************************************************
2122                                    ;TEST DIV INSTRUCTION
2123                                    ;;*****************************************************************
2124  005476  012701  125252   DIV0:    MOV     #125252,R1      ;LOAD LOW ORDER DIVIDEND
2125  005502  012700  177777            MOV     #-1,R0          ;HIGH ORDER DIVIDEND
2126  005506  000261                    SEC                     ;SET C BIT TO 1
2127  005510  071027  000002            DIV     #2,R0           ;R1:R0=177777,125252/2
2128  005514  103403                    BCS     1$              ;BR IF C BIT ERROR
2129  005516  102402                    BVS     1$              ;BR IF V BIT ERROR
2130  005520  001401                    BEQ     1$              ;BR IF Z BIT ERROR
2131  005522  100401                    BMI     2$              ;BR IF N BIT OK
2132  005524  104126   1$:      ERROR   126             ;INCORRECT DONDITION CODE FOR ''DIV''
2133  005526  005701   2$:      TST     R1              ;CHECK LOW ORDER RESULT
2134  005530  001003                    BNE     3$              ;BR IF ERROR
2135  005532  020027  152525            CMP     R0,#152525      ;CHECK HIGH ORDER RESULT
2136  005536  001401                    BEQ     FIS             ;BR IF OK
2137  005540  104127   3$:      ERROR   127             ;INCORRECT DATA FOR ''DIV''
```

```
2138                            ;;*****************************************************************
2139                            ;TEST ALL FLOATING INSTRUCTIONS TOGETHER
2140                            ;                                          045670,123456
2141                            ;       134343,107070 * 032107,065432 * ------------------------------
2142                            ;                                       (135252,125252 - 040616,016161)
2143                            ;
2144                            ;       THE RESULT =  137201,115230
2145                            ;;*****************************************************************
2146  005542  012704  032276   FIS:    MOV     #LSTAD,R4               ;LOAD R4 STACK POINTER
2147  005546  012744  107070           MOV     #107070,-(R4)           ;LOAD DATA ONTO STACK
2148  005552  012744  134343           MOV     #134343,-(R4)
2149  005556  012744  065432           MOV     #065432,-(R4)
2150  005562  012744  032107           MOV     #032107,-(R4)
2151  005566  012744  123456           MOV     #123456,-(R4)
2152  005572  012744  045670           MOV     #045670,-(R4)
2153  005576  012744  125252           MOV     #125252,-(R4)
2154  005602  012744  135252           MOV     #135252,-(R4)
2155  005606  012744  016161           MOV     #016161,-(R4)
2156  005612  012744  040616           MOV     #040616,-(R4)
2157  005616  000257                   CCC
2158  005620  000264                   SEZ                             ;SET Z BIT
2159  005622  000240                   NOP
2160  005624  075014                   FSUB    R4      ;135252,125252-040616,016161=140616,017434
2161  005626  075034                   FDIV    R4      ;045670,123456/140616,017434=145246,047065
2162  005630  075024                   FMUL    R4      ;032107,065432*145246,047065=137201,106137
2163  005632  075004                   FADD    R4      ;134343,107070+137201,106137=137201,115230
2164  005634  103403                   BCS     1$              ;BR IF C SET, REPORT ERROR
2165  005636  102402                   BVS     1$              ;BR IF V SET, REPORT ERROR
2166  005640  001401                   BEQ     1$              ;BR IF Z SET, REPORT ERROR
2167  005642  100401                   BMI     2$              ;BR IF N SET, DO NEXT TEST
2168  005644  104126           1$:     ERROR   126             ;INCORRECT CONDITION CODE AFTER ''FADD,FSUB,FMUL,FDIV''
2169  005646  012437  005724   2$:     MOV     (R4)+,ANS1      ;SAVE FIRST HALF OF THE ANSWER
2170  005652  012437  005726           MOV     (R4)+,ANS2      ;SAVE 2ND HALF
2171  005656  020427  032276           CMP     R4,#LSTAD       ;CHECK R4 STACK POINTER
2172  005662  001401                   BEQ     3$              ;BR IF CORRECT
2173  005664  104127                   ERROR   127             ;INCORRECT STACK POINTER VALUE
2174  005666  022737  137201  005724 3$:     CMP     #137201,ANS1    ;TEST 1ST HALF OF RESULT
2175  005674  001401                   BEQ     4$              ;BR IF CORRECT
2176  005676  104127                   ERROR   127             ;INCORRECT DATA FROM A ''FIS INSTRUCTION''
2177  005700  022737  115230  005726 4$:     CMP     #115230,ANS2    ;TEST 2ND HALF OF RESULT
2178  005706  001401                   BEQ     5$              ;BR IF CORRECT
2179  005710  104127                   ERROR   127             ;INCORRECT DATA FORM A ''FIS INSTRUCTION''
2180  005712  005337  005730   5$:     DEC     EISCNT          ;FINISHED THE # OF EXECUTION LOOPS ?
2181  005716  001405                   BEQ     EISDON          ;BR IF FINISHED
2182  005720  000137  005106           JMP     ASH1            ;LOOP UNTIL DONE
2183  005724  000000           ANS1:   0
2184  005726  000000           ANS2:   0
2185  005730  000000           EISCNT: 0
2186
2187  005732  004737  012624   EISDON: JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2188  005736  000410                       BR  NOEIS           ;RET HERE IF NOT
2189  005740  032777  010000  173172       BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2190  005746  001404                   BEQ     NOEIS           ;BR IF CLEARED
2191  005750  104401  022542           TYPE    ,FISEIS         ;REPORT EIS/FIS TESTING COMPLETED
2192  005754  104401  022574           TYPE    ,DUN            ;REPORT ''DONE''
2193  005760                   NOEIS:                  ;TAG TO BYPASS THE EIS/FIS TEST SECTION
```

```
2194
2195                                    ;;*****************************************************************
2196                                    ;       START LINE CLOCK
2197                                    ;;*****************************************************************
2198
2199  005760  032737  000200  001246  CLKTST: BIT     #BIT7,$DEVM     ;DROP TEST?
2200  005766  001034                          BNE     PRTST           ;BR IF YES
2201  005770  012737  014174  000100          MOV     #INTSRV,CLKVEC  ;SETUP VECTOR AREA
2202  005776  012737  000200  000102          MOV     #200,CLKVEC+2   ;LOCKOUT OTHER INTER.
2203  006004  005037  014202                  CLR     INTFLG
2204  006010  052737  000100  177546          BIS     #IE,CLK         ;SET CLOCK LOOSE!
2205
2206  006016  004537  013340                  JSR     R5,TIMER        ;SEE IF INTFLG GOES TO 100
2207  006022  014202                            INTFLG
2208  006024  000100                            100
2209  006026  104032                          ERROR   32              ;CLK NOT RESPONDING
2210  006030  000413                          BR      PRTST
2211
2212  006032  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2213  006036  000410                              BR  PRTST           ;RET HERE IF NOT
2214  006040  032777  010000  173072          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2215  006046  001404                          BEQ     PRTST
2216  006050  104401  021676                  TYPE    ,CLOCK
2217  006054  104401  022467                  TYPE    ,RUN
```

```
2218
2219                                    ;;*****************************************************************
2220                                    ;       START PRINTER
2221                                    ;;*****************************************************************
2222
2223  006060  032737  100000  001246  PRTST:  BIT     #BIT15,$DEVM    ;DROP TEST?
2224  006066  001067                          BNE     SMTST           ;BR IF YES
2225  006070  005737  012616                  TST     PRPRES          ;PRINTER THERE?
2226  006074  001464                          BEQ     SMTST           ;BR IF NO
2227
2228  006076  013737  003520  003534  2$:     MOV     PRBAUD,HOLD
2229  006104  052737  020014  003534          BIS     #<PRT!CHAR8>,HOLD
2230  006112  013737  003534  177420          MOV     HOLD,PARAM      ;SETUP PARAM REG
2231
2232  006120  012737  014204  000200          MOV     #PRSRV,PRVEC    ;ELSE SETUP PRINTER VECTOR
2233  006126  012737  000200  000202          MOV     #200,PRVEC+2    ;LOCKOUT INTERR
2234
2235  006134  005037  014332                  CLR     LINCT           ;CLEAR LINE CTR
2236  006140  012737  000015  014330          MOV     #CR,PRCHR       ;CHAR TO BE PRINTED
2237                                                                  ;'DEL' (177) CAN BE USED TO CLR PRINTER BUFF
2238  006146  052737  000100  177514          BIS     #IE,PRS         ;SET PRINTER LOOSE!
2239                                                                  ;WILL PRINT CONTINUOUS 132 COLUMN LINES
2240                                                                  ;EA BEGINNING WITH ASCII 40 THRU 176
2241  006154  004537  013340                  JSR     R5,TIMER        ;SEE IF LINCT GOES TO 5
2242  006160  014332                            LINCT
2243  006162  000005                            5
2244  006164  104033                          ERROR   33              ;PRINTER NOT RESPONDING
2245  006166  000427                          BR      SMTST
2246
2247  006170  005737  012620                  TST     PRPORT          ;EXT LOOPBACK MODE?
2248  006174  001403                          BEQ     3$              ;BR IF NO
2249  006176  042737  000100  177514          BIC     #IE,PRS         ;ELSE SHUT DOWN PRINTER
2250
2251  006204  004737  012624          3$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2252  006210  000416                              BR  SMTST           ;RET HERE IF NOT
2253  006212  032777  010000  172720          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2254  006220  001412                          BEQ     SMTST
2255  006222  104401  021544                  TYPE    ,PRINT
2256  006226  005737  012620                  TST     PRPORT          ;EXT LOOPBACK MODE?
2257  006232  001403                          BEQ     4$              ;BR IF NO
2258  006234  104401  021555                  TYPE    ,PORT
2259  006240  000402                          BR      SMTST
2260  006242  104401  022467          4$:     TYPE    ,RUN
```

```
2261
2262                                    ;;*****************************************************************
2263                                    ;       START SYNC COMM PORT
2264                                    ;;*****************************************************************
2265
2266  006246  032737  002000  001246  SMTST:  BIT     #BIT10,$DEVM    ;DROP TEST?
2267  006254  001402                          BEQ     8$              ;BR IF NO
2268  006256  000137  006562                  JMP     AMTST           ;ELSE JUMP TEST
2269
2270  006262  123727  001210  000001  8$:     CMPB    $ENV,#1         ;UNDER APT?
2271  006270  001003                          BNE     9$              ;BR IF NO
2272  006272  005737  001176                  TST     $PASS           ;ELSE 1'ST PASS?
2273  006276  001131                          BNE     AMTST           ;EXIT TEST IF NO
2274
2275  006300  005037  015024          9$:     CLR     LSTCHR          ;LAST CHAR FLAG FOR XMIT
2276  006304  005237  003514                  INC     SMFLG           ;SET SYNC MODEM FLAG
2277  006310  013737  003522  003534          MOV     SMBAUD,HOLD
2278  006316  052737  030000  003534          BIS     #SMOD,HOLD
2279  006324  013737  003534  177420          MOV     HOLD,PARAM      ;SET PARAM REG
2280  006332  052737  000400  176624          BIS     #MR,SMXC        ;MASTER RESET
2281  006340  052737  004000  176624          BIS     #MCLK,SMXC      ;MAINT CLOCK
2282
2283  006346  032737  000010  001246          BIT     #BIT3,$DEVM                     ;EXT LOOPBACK?
2284  006354  001403                          BEQ     1$                              ;BR IF YES
2285  006356  052737  014000  176624          BIS     #<MM!MCLK>,SMXC                 ;ELSE SET MAINT MODE FOR INT. LOOPBACK
2286  006364  012737  007026  176622  1$:     MOV     #<CHR8!ODDPAR!026>,SMPAR        ;SETUP SYNC PARAMETER REGISTER
2287  006372  052737  000026  176620          BIS     #<RTS!DTR!SRSYN>,SMRC           ;SET REQ TO SEND, DATA TERM RDY
2288                                                                                  ;& SEARCH SYNC
2289  006400  052737  000020  176624          BIS     #SEND,SMXC      ;ENABLE XMIT SEND
2290  006406  012737  000026  176626          MOV     #026,SMXB       ;XMIT SYNC CHAR
2291
2292  006414  004537  013632                  JSR     R5,TIMER2       ;WAIT FOR DONE
2293  006420  176620                              SMRC
2294  006422  000200                              DONE
2295  006424  104045                          ERROR   45              ;NO DONE
2296  006426  000435                          BR      5$              ;EXIT
2297
2298  006430  013737  176622  014734          MOV     SMRB,COMHLD     ;READ WORD
2299  006436  023727  014734  000026          CMP     COMHLD,#026     ;IS IT SYNC CHAR?
2300  006444  001402                          BEQ     4$              ;BR IF YES
2301  006446  104006                          ERROR   6               ;DID NOT REC SYNC CHAR
2302  006450  000424                          BR      5$              ;EXIT
2303
2304  006452  004537  013314          4$:     JSR     R5,SETVEC       ;SETUP VECTOR AREA
2305  006456  000340                              SMRVEC
2306  006460  015026                              SMRSRV
2307  006462  014736                              SMXSRV
```

```
2308
2309  006464  012737  000027  014732          MOV     #027,COMCHR     ;1'ST CHAR TO BE XMITTED
2310  006472  052737  000100  176620          BIS     #IE,SMRC        ;SET SYNC MODEM LOOSE!
2311  006500  052737  000100  176624          BIS     #IE,SMXC
2312
2313  006506  004537  013340                  JSR     R5,TIMER        ;SEE IF COMCHR GOES TO ALL 1'S (DONE)
2314  006512  014732                            COMCHR
2315  006514  177777                            177777
2316  006516  104037                          ERROR   37              ;SYNC COMM NOT RESPONDING
2317  006520  000240                          NOP
2318
2319  006522  004737  012624          5$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2320  006526  000413                              BR  2$              ;RET HERE IF NOT
2321  006530  012737  000400  176624          MOV     #MR,SMXC        ;MASTER RESET
2322  006536  032777  010000  172374          BIT     #BIT12,@SWR     ;DO TYPEOUT IF SET
2323  006544  001404                          BEQ     2$              ;EXIT IF NOT
2324  006546  104401  021515                  TYPE    ,SCOM
2325  006552  104401  022574                  TYPE    ,DUN
2326  006556  005037  003514          2$:     CLR     SMFLG           ;CLEAR FLAG
```

```
                                        ;;*****************************************************************
                                        ;       START ASYNC COMM PORT
                                        ;;*****************************************************************

006562  032737  004000  001246  AMTST:  BIT     #BIT11,$DEVM    ;DROP TEST?
006570  001065                          BNE     CL1TST          ;BR IF YES
006572  013737  003524  003534          MOV     AMBAUD,HOLD

006600  032737  000010  001246          BIT     #BIT3,$DEVM                     ;EXT LOOPBACK?
006606  001404                          BEQ     1$                              ;BR IF YES
006610  052737  030036  003534          BIS     #<AMOD!OPAR!CHAR8!MAINT>,HOLD   ;INT LOOPBACK
006616  000403                          BR      3$
006620  052737  030034  003534  1$:     BIS     #<AMOD!OPAR!CHAR8>,HOLD         ;EXT LOOPBACK
006626  013737  003534  177420  3$:     MOV     HOLD,PARAM                      ;SET PARAM REG
006634  052737  000006  176610          BIS     #<RTS!DTR>,AMRC                 ;SET REQ TO SEND & DATA TERM RDY

006642  004537  013314                  JSR     R5,SETVEC       ;SETUP VECTOR AREA
006646  000330                              AMRVEC
006650  014654                              AMRSRV
006652  014636                              AMXSRV

006654  005037  014732                  CLR     COMCHR          ;CHAR TO BE XMITTED
006660  013737  176612  014734          MOV     AMRB,COMHLD     ;CLR OUT ANY PREVIOUS STORED WORD
006666  052737  000100  176610          BIS     #IE,AMRC        ;SET ASYNC MODEM LOOSE!
006674  052737  000100  176614          BIS     #IE,AMXC

006702  004537  013340                  JSR     R5,TIMER        ;SEE IF COMCHR GOES TO 377
006706  014732                            COMCHR
006710  000377                            377
006712  104040                          ERROR   40              ;ASYNC COMM NOT RESPONDING
006714  000413                          BR      CL1TST

006716  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
006722  000410                              BR  CL1TST          ;RET HERE IF NOT
006724  032777  010000  172206          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
006732  001404                          BEQ     CL1TST
006734  104401  021530                  TYPE    ,ACOM
006740  104401  022467                  TYPE    ,RUN
```

```
2366
2367                                    ;;*****************************************************************
2368                                    ;       START CLUSTER TERMINAL #1
2369                                    ;;*****************************************************************
2370
2371  006744  005737  012614           CL1TST: TST     CLPRES          ;OPTION PRESENT?
2372  006750  001002                           BNE     1$              ;BR IF YES
2373  006752  000137  007506                   JMP     DXTST0
2374
2375  006756  032737  004000  001246  1$:     BIT     #4000,$DEVM     ;DROP ASYNC?
2376  006764  001406                           BEQ     4$              ;BR IF NO
2377  006766  042737  000006  176610           BIC     #<RTS!DTR>,AMRC ;MUST BE TOGGLED
2378  006774  052737  000006  176610           BIS     #<RTS!DTR>,AMRC ;PRIME IT
2379
2380  007002  032737  010000  001246  4$:     BIT     #BIT12,$DEVM    ;DROP TEST?
2381  007010  001062                           BNE     CL2TST          ;BR IF YES
2382  007012  013737  003526  003534           MOV     C1BAUD,HOLD
2383
2384  007020  032737  000001  001246           BIT     #BIT0,$DEVM             ;TERM #1 SET FOR EXT LOOPBACK?
2385  007026  001404                           BEQ     2$                      ;BR IF YES
2386  007030  052737  000016  003534           BIS     #<CT1!CHAR8!MAINT>,HOLD ;INT LOOPBACK
2387  007036  000403                           BR      3$
2388  007040  052737  000014  003534  2$:     BIS     #<CT1!CHAR8>,HOLD       ;EXT LOOPBACK
2389
2390  007046  013737  003534  177420  3$:     MOV     HOLD,PARAM      ;SET PARAM REG
2391  007054  004537  013314                   JSR     R5,SETVEC       ;SETUP INTERR VECTORS
2392  007060  000300                               TK1VEC
2393  007062  014354                               TK1SRV
2394  007064  014336                               TP1SRV
2395
2396  007066  005037  014432                   CLR     T1CHR           ;CHAR TO BE XMITTED
2397  007072  013737  176502  014434           MOV     TK1B,T1HLD      ;READ CHAR (CLEAR STALE DATA)
2398  007100  052737  000100  176500           BIS     #IE,TK1S        ;SET CLUSTER TERM #1 LOOSE!
2399  007106  052737  000100  176504           BIS     #IE,TP1S
2400
2401  007114  004537  013340                   JSR     R5,TIMER        ;SEE IF T1CHR GOES TO 377
2402  007120  014432                             T1CHR
2403  007122  000377                             377
2404  007124  104034                           ERROR   34              ;CLUSTER TERM 1 NOT RESPONDING
2405  007126  000413                           BR      CL2TST
2406
2407  007130  004737  012624                   JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2408  007134  000410                               BR  CL2TST          ;RET HERE IF NOT
2409  007136  032777  010000  171774           BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2410  007144  001404                           BEQ     CL2TST
2411  007146  104401  021462                   TYPE    ,CLT1
2412  007152  104401  022467                   TYPE    ,RUN
```

```
2413
2414                                    ;;*****************************************************************
2415                                    ;       START CLUSTER TERMINAL #2
2416                                    ;;*****************************************************************
2417
2418  007156  032737  020000  001246  CL2TST: BIT     #BIT13,$DEVM    ;DROP TEST?
2419  007164  001062                         BNE     CL3TST          ;BR IF YES
2420  007166  013737  003530  003534         MOV     C2BAUD,HOLD
2421
2422  007174  032737  000002  001246         BIT     #BIT1,$DEVM             ;TERM #2 SET FOR EXT LOOPBACK?
2423  007202  001404                         BEQ     1$                      ;BR IF YES
2424  007204  052737  050016  003534         BIS     #<CT2!CHAR8!MAINT>,HOLD ;INT LOOPBACK
2425  007212  000403                         BR      2$
2426  007214  052737  050014  003534  1$:    BIS     #<CT2!CHAR8>,HOLD       ;EXT LOOPBACK
2427
2428  007222  013737  003534  177420  2$:    MOV     HOLD,PARAM      ;SET PARAM REG
2429  007230  004537  013314                 JSR     R5,SETVEC       ;SETUP INTERR VECTORS
2430  007234  000310                             TK2VEC
2431  007236  014454                             TK2SRV
2432  007240  014436                             TP2SRV
2433
2434  007242  005037  014532                 CLR     T2CHR           ;CHAR TO BE XMITTED
2435  007246  013737  176512  014534         MOV     TK2B,T2HLD      ;READ CHAR (CLEAR STALE DATA)
2436  007254  052737  000100  176510         BIS     #IE,TK2S        ;SET CLUSTER TERM #2 LOOSE!
2437  007262  052737  000100  176514         BIS     #IE,TP2S
2438
2439  007270  004537  013340                 JSR     R5,TIMER        ;SEE IF T2CHR GOES TO 377
2440  007274  014532                           T2CHR
2441  007276  000377                           377
2442  007300  104035                         ERROR   35              ;CLUSTER TERM 2 NOT RESPONDING
2443  007302  000413                         BR      CL3TST
2444
2445  007304  004737  012624                 JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2446  007310  000410                             BR  CL3TST          ;RET HERE IF NOT
2447  007312  032777  010000  171620         BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2448  007320  001404                         BEQ     CL3TST
2449  007322  104401  021473                 TYPE    ,CLT2
2450  007326  104401  022467                 TYPE    ,RUN
```

```
2451
2452                                    ;;*****************************************************************
2453                                    ;       START CLUSTER TERMINAL #3
2454                                    ;;*****************************************************************
2455
2456  007332  032737  040000  001246  CL3TST: BIT     #BIT14,$DEVM    ;DROP TEST?
2457  007340  001062                          BNE     DXTSTO          ;BR IF YES
2458  007342  013737  003532  003534          MOV     C3BAUD,HOLD
2459
2460  007350  032737  000004  001246          BIT     #BIT2,$DEVM             ;TERM #3 SET FOR EXT LOOPBACK?
2461  007356  001404                          BEQ     1$                      ;BR IF YES
2462  007360  052737  060016  003534          BIS     #<CT3!CHAR8!MAINT>,HOLD ;INT LOOPBACK
2463  007366  000403                          BR      2$
2464  007370  052737  060014  003534  1$:     BIS     #<CT3!CHAR8>,HOLD       ;EXT LOOPBACK
2465
2466  007376  013737  003534  177420  2$:     MOV     HOLD,PARAM      ;SET PARAM REG
2467  007404  004537  013314                  JSR     R5,SETVEC       ;SETUP INTERR VECTORS
2468  007410  000320                              TK3VEC
2469  007412  014554                              TK3SRV
2470  007414  014536                              TP3SRV
2471
2472  007416  005037  014632                  CLR     T3CHR           ;CHAR TO BE XMITTED
2473  007422  013737  176522  014634          MOV     TK3B,T3HLD      ;READ CHAR (CLEAR STALE DATA)
2474  007430  052737  000100  176520          BIS     #IE,TK3S        ;SET CLUSTER TERM #3 LOOSE!
2475  007436  052737  000100  176524          BIS     #IE,TP3S
2476
2477  007444  004537  013340                  JSR     R5,TIMER        ;SEE IF T3CHR GOES TO 377
2478  007450  014632                            T3CHR
2479  007452  000377                            377
2480  007454  104036                          ERROR   36              ;CLUSTER TERM 3 NOT RESPONDING
2481  007456  000413                          BR      DXTSTO
2482
2483  007460  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2484  007464  000410                              BR  DXTSTO          ;RET HERE IF NOT
2485  007466  032777  010000  171444          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
2486  007474  001404                          BEQ     DXTSTO
2487  007476  104401  021504                  TYPE    ,CLT3
2488  007502  104401  022467                  TYPE    ,RUN
```

```
2489
2490                                    ;;*****************************************************************
2491                                    ;       FILL & EMPTY BUFFER TEST:  DRIVE 0
2492                                    ;
2493                                    ;A FILL BUFFER COMMAND IS ISSUED WITH ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
2494                                    ;AN EMPTY BUFFER COMMAND IS ISSUED TO FILL 'DOBUF'.
2495                                    ;UPON COMPLETION, DOBUF IS CHECKED FOR ALTERNATING WORDS OF ALL 0'S & ALL 1'S.
2496                                    ;ERRORS IN THIS TEST MAY INDICATE CABLE CROSS-TALK INTERFERENCE.
2497                                    ;;*****************************************************************
2498
2499  007506  005037  003476           DXTST0: CLR     DXTST           ;0 = DXTST0
2500
2501  007512  032737  000040  001246           BIT     #BIT5,$DEVM     ;DROP BUFFER TESTS?
2502  007520  001113                           BNE     DXTST1          ;BR IF YES
2503
2504  007522  004537  013632                   JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2505  007526  177170                              RXCS
2506  007530  000200                              DONE
2507  007532  104075                           ERROR   75              ;NO DONE
2508  007534  000500                           BR      8$              ;EXIT
2509
2510  007536  012737  000011  177170           MOV     #INITAL,RXCS    ;DO AN INITIALIZE COMMAND
2511
2512  007544  004537  013632                   JSR     R5,TIMER2       ;WAIT FOR DONE
2513  007550  177170                              RXCS
2514  007552  000200                              DONE
2515  007554  104114                           ERROR   114             ;NO DONE
2516  007556  000467                           BR      8$
2517
2518  007560  005737  177170                   TST     RXCS            ;ERROR?
2519  007564  100001                           BPL     6$              ;BR IF NO
2520  007566  104130                           ERROR   130             ;INIT CMD ERROR
2521
2522                                                                   ;FILL BUFFER
2523  007570  012700  000100           6$:     MOV     #100,R0         ;WORD CT
2524  007574  005001                           CLR     R1              ;WORD
2525  007576  012737  000001  177170           MOV     #FBUF,RXCS      ;ISSUE FILL BUFF COMMAND
2526  007604  010137  177172           1$:     MOV     R1,RXDB
2527  007610  005101                           COM     R1              ;COMPLEMENT WORD
2528  007612  005300                           DEC     R0              ;DONE ALL 100 WORDS?
2529  007614  001373                           BNE     1$              ;BR IF NO
2530
2531  007616  004537  013632                   JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2532  007622  177170                              RXCS
2533  007624  000200                              DONE
2534  007626  104115                           ERROR   115             ;NO DONE
2535  007630  000442                           BR      8$              ;EXIT
```

```
2536
2537  007632  012700  000100              MOV     #100,R0         ;WORD CT
2538  007636  012701  002604              MOV     #DOBUF,R1       ;BUFFER ADDR
2539  007642  012737  000003  177170      MOV     #EBUF,RXCS      ;ISSUE EMPTY BUFF COMMAND
2540  007650  013721  177172      2$:     MOV     RXDB,(R1)+      ;STORE WORD
2541  007654  005300                      DEC     R0              ;DONE?
2542  007656  001374                      BNE     2$              ;BR IF NO
2543
2544  007660  004537  013632              JSR     R5,TIMER2       ;WAIT FOR DISK DONE
2545  007664  177170                          RXCS
2546  007666  000200                          DONE
2547  007670  104116                      ERROR   116             ;NO DONE
2548  007672  000421                      BR      8$              ;EXIT
2549
2550                                              ;CHECK IT
2551  007674  012701  002604              MOV     #DOBUF,R1       ;BUFFER ADDR
2552  007700  012137  007742      3$:     MOV     (R1)+,EBHLD     ;STORE IT
2553  007704  001401                      BEQ     4$              ;BR IF ZERO
2554  007706  104076                      ERROR   76              ;WORD NOT ALL ZEROS
2555
2556  007710  012137  007742      4$:     MOV     (R1)+,EBHLD
2557  007714  023727  007742  177777      CMP     EBHLD,#-1       ;ALL 1'S?
2558  007722  001401                      BEQ     5$              ;BR IF YES
2559  007724  104077                      ERROR   77              ;WORD NOT ALL ONES
2560
2561  007726  020127  003004      5$:     CMP     R1,#DOBUF+200   ;END OF DOBUFF?
2562  007732  001362                      BNE     3$              ;BR IF NO
2563  007734  000405                      BR      DXTST1          ;GO TO NXT TST
2564
2565  007736  000137  011706      8$:     JMP     EOP             ;EXIT ALL DISK TESTS
2566
2567  007742  000000          EBHLD:  0               ;STORAGE        FOR ERROR REPORT
2568  007744  000000          EXP0:   0               ;EXPECT 0'S     FOR ERROR REPORT
2569  007746  177777          EXP1:   -1              ;EXPECT 1'S     FOR ERROR REPORT
```

```
2570
2571                                    ;;*****************************************************************
2572                                    ;       START DX0 & DX1 DATA PATTERN
2573                                    ;
2574                                    ;EACH SECTOR IS FILLED WITH ITS TRACK & SECTOR ADDRESS AS DATA.
2575                                    ;DX1 WILL ALWAYS HAVE ITS DATA BIT 15 SET TO DISTINGUISH IT FROM DX0.
2576                                    ;EACH TRACK IS READ & CHECKED BEFORE GOING TO THE NEXT TRACK.
2577                                    ;DATA IS WRITTEN ON ALTERANTE SECTORS TO AVOID LATENCY TIMES.
2578                                    ;IE: SECTORS 1,3,5...31 FOLLOWED BY 2,4,6...32.
2579                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2580                                    ;1'ST PASS PERFORMED ON TRACKS 0-4, 40-44, 72-76(10)
2581                                    ;SUBSEQUENT PASSES ON ALL TRACKS
2582                                    ;;*****************************************************************
2583
2584  007750  012737  000001  003476  DXTST1: MOV     #1,DXTST        ;1 = DATA PATT TESTS
2585
2586  007756  005037  002566          CLR     D0TRK           ;STARTING TRACK
2587  007762  005037  003036          CLR     D1TRK
2588
2589  007766  005037  002560          12$:    CLR     D0PAT           ;INIT DX0 FLAGS
2590  007772  005037  002564          CLR     D0ERR
2591  007776  005037  002574          CLR     D0CERR
2592  010002  012737  000001  002570  MOV     #1,D0SEC
2593
2594  010010  005037  003030          CLR     D1PAT           ;INIT DX1 FLAGS
2595  010014  005037  003034          CLR     D1ERR
2596  010020  005037  003044          CLR     D1CERR
2597  010024  012737  000001  003040  MOV     #1,D1SEC
2598
2599  010032  032737  000400  001246  BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2600  010040  001031                  BNE     2$              ;BR IF YES
2601
2602  010042  012737  015126  000264  MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2603  010050  012737  000200  000266  MOV     #200,RXVEC+2
2604
2605  010056  005737  003512  1$:     TST     COMP2           ;DOING COMPAT PASS 2?
2606  010062  001404                  BEQ     10$             ;BR IF NO
2607  010064  012737  015632  015226  MOV     #D0RSEC,DODISP  ;ELSE DO READS ONLY
2608  010072  000403                  BR      7$
2609
2610  010074  012737  015244  015226  10$:    MOV     #D0FBUF,DODISP  ;DO NORMAL TESTING
2611  010102  005037  003500  7$:      CLR     FIN             ;DO DX1 WHEN SET
2612  010106  052737  000100  177170  BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE!
2613
2614  010114  004737  013424          JSR     PC,TIMDO
2615  010120  104041                  ERROR   41              ;DX0 NOT GETTING TRACK DONE FLAG
2616  010122  000240                  NOP
2617
2618  010124  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2619  010132  001031                  BNE     4$              ;BR IF YES
2620
2621  010134  012737  015126  000264  MOV     #RXSRV,RXVEC    ;SETUP VECTOR AREA
2622  010142  012737  000200  000266  MOV     #200,RXVEC+2
```

```
2623
2624  010150  005737  003512          3$:     TST     COMP2           ;DOING COMPAT PASS 2?
2625  010154  001404                          BEQ     11$             ;BR IF NO
2626  010156  012737  017470  015234          MOV     #D1RSEC,D1DISP  ;ELSE DO READ ONLY
2627  010164  000403                          BR      8$
2628
2629  010166  012737  017102  015234  11$:    MOV     #D1FBUF,D1DISP  ;DO NORMAL TESTING
2630  010174  005037  003500          8$:     CLR     FIN             ;GO BACK TO DX0 WHEN SET
2631  010200  052737  000120  177170          BIS     #<IE.DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2632
2633  010206  004737  013524                  JSR     PC,TIMD1
2634  010212  104042                          ERROR   42              ;DX1 NOT GETTING TRACK DONE FLAG
2635  010214  000240                          NOP
2636
2637  010216  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2638  010224  001007                          BNE     5$              ;BR IF YES
2639  010226  005737  002560                  TST     DOPAT           ;ELSE IS DX0 ALL DONE?
2640  010232  001004                          BNE     5$              ;BR IF YES
2641  010234  005737  013522                  TST     DOTMO           ;TIMEOUT?
2642  010240  001706                          BEQ     1$              ;BR IF NO
2643  010242  000717                          BR      7$              ;ELSE CONT LAST COMMAND
2644
2645  010244  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2646  010252  001007                          BNE     6$              ;BR IF YES
2647  010254  005737  003030                  TST     D1PAT           ;ELSE IS DX1 ALL DONE?
2648  010260  001004                          BNE     6$              ;BR IF YES
2649  010262  005737  013616                  TST     D1TMO           ;TIMEOUT?
2650  010266  001730                          BEQ     3$              ;BR IF NO
2651  010270  000741                          BR      8$              ;ELSE CONT LAST COMMAND
2652
2653  010272  005737  003510          6$:     TST     COMP1           ;DOING COMPAT PASS 1?
2654  010276  001402                          BEQ     13$             ;BR IF NO
2655  010300  000137  011706                  JMP     EOP             ;ELSE ALL DONE
2656
2657  010304  005737  001176          13$:    TST     $PASS           ;1'ST PASS?
2658  010310  001035                          BNE     DXTST2          ;BR IF NO
2659
2660  010312  023727  003502  000004          CMP     MAXTRK,#4       ;JUST DID 5 OUTER TRACKS?
2661  010320  001012                          BNE     14$             ;BR IF NO
2662  010322  012737  000050  002566          MOV     #40.,DOTRK      ;ELSE  SETUP FOR 5 MIDDLE TRKS
2663  010330  012737  000050  003036          MOV     #40.,D1TRK
2664  010336  012737  000054  003502          MOV     #44.,MAXTRK
2665  010344  000610                          BR      12$             ;GO DO IT!
2666
2667  010346  023727  003502  000054  14$.    CMP     MAXTRK,#44.     ;JUST DID 5 MIDDLE TRACKS?
2668  010354  001013                          BNE     DXTST2          ;BR IF NO, MUST BE ALL DONE
2669  010356  012737  000110  002566          MOV     #72.,DOTRK      ;ELSE SETUP FOR 5 INNER TRKS
2670  010364  012737  000110  003036          MOV     #72.,D1TRK
2671  010372  012737  000114  003502          MOV     #76.,MAXTRK
2672  010400  000137  007766                  JMP     12$             ;DO IT!
2673
```

```
2674
2675                                    ;;*****************************************************************
2676                                    ;       START DX0 & DX1 RANDOM SEEKS
2677                                    ;
2678                                    ;RANDOM SEEKS ARE PERFORMED ON EACH DISK.
2679                                    ;DATA IS READ & VERIFIED TO BE CORRECT FROM THE PREVIOUS TEST.
2680                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2681                                    ;1'ST PASS PERFORMS 20 RANDOM SEEKS ON 1'ST 5 TRACKS.
2682                                    ;SUBSEQUENT PASSES PERFORMS 500 RANDOM SEEKS BETWEEN ALL TRACKS.
2683                                    ;;*****************************************************************
2684
2685  010404  012737  000002  003476  DXTST2: MOV     #2,DXTST        ;2 = RANDOM SEEK TESTS
2686  010412  005737  001176                  TST     $PASS           ;1'ST PASS?
2687  010416  001003                          BNE     7$              ;BR IF NO
2688  010420  012737  000004  003502          MOV     #4,MAXTRK       ;ELSE JUST DO TRKS 0-4
2689
2690  010426  005037  002564          7$:     CLR     DOERR           ;INIT DX0 FLAGS
2691  010432  005037  002574                  CLR     DOCERR
2692  010436  005037  002562                  CLR     DOCNT
2693
2694  010442  005037  003034                  CLR     D1ERR           ;INIT DX1 FLAGS
2695  010446  005037  003044                  CLR     D1CERR
2696  010452  005037  003032                  CLR     D1CNT
2697
2698  010456  032737  000400  001246          BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2699  010464  001041                          BNE     2$              ;BR IF YES
2700
2701  010466  005037  013074          1$:     CLR     LOLIM
2702  010472  013737  003502  013076          MOV     MAXTRK,HILIM
2703  010500  004737  012754                  JSR     PC,RAND
2704  010504  010137  002566                  MOV     R1,DOTRK        ;GET RANDOM TRACK #
2705
2706  010510  005237  013074                  INC     LOLIM
2707  010514  012737  000032  013076          MOV     #32,HILIM
2708  010522  004737  012754                  JSR     PC,RAND
2709  010526  010137  002570                  MOV     R1,DOSEC        ;GET RANDOM SECTOR #
2710
2711  010532  004537  014006                  JSR     R5,CKOBAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
2712  010536  000753                             BR   1$              ;BR IF RETURN HERE
2713
2714  010540  012737  015632  015226          MOV     #DORSEC,DODISP  ;SET DISPATCH TABLE TO POINT TO
2715                                                                  ;READ SECTOR HANDLER.
2716  010546  005037  003500                  CLR     FIN             ;DO DX1 WHEN SET
2717  010552  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERRUPTS LOOSE.
2718
2719  010560  004737  013424                  JSR     PC,TIMDO
2720  010564  104117                          ERROR   117             ;DX0 NOT GETTING DONE FLAG
2721  010566  000240                          NOP
```

```
2722
2723  010570  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2724  010576  001041                          BNE     4$              ;BR IF YES
2725
2726  010600  005037  013074          3$:     CLR     LOLIM
2727  010604  013737  003502  013076          MOV     MAXTRK,HILIM
2728  010612  004737  012754                  JSR     PC,RAND
2729  010616  010137  003036                  MOV     R1,D1TRK        ;GET RANDOM TRACK #
2730
2731  010622  005237  013074                  INC     LOLIM
2732  010626  012737  000032  013076          MOV     #32,HILIM
2733  010634  004737  012754                  JSR     PC,RAND
2734  010640  010137  003040                  MOV     R1,D1SEC        ;GET RANDOM SECTOR #
2735
2736  010644  004537  014120                  JSR     R5,CK1BAD       ;SEE IF THIS TRK/SEC FLAGGED BAD
2737  010650  000753                             BR   3$              ;BR IF RETURN HERE
2738
2739  010652  012737  017470  015234          MOV     #D1RSEC,D1DISP  ;SET DISPATCH TABLE TO POINT TO
2740                                                                  ;READ SECTOR HANDLER.
2741  010660  005037  003500                  CLR     FIN             ;GO BACK TO DX0 WHEN SET
2742  010664  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERRUPTS LOOSE!
2743
2744  010672  004737  013524                  JSR     PC,TIMD1
2745  010676  104120                          ERROR   120             ;DX1 NOT GETTING DONE FLAG
2746  010700  000240                          NOP
2747
2748  010702  032737  000400  001246  4$:     BIT     #BIT8,$DEVM     ;DROP DX0?
2749  010710  001004                          BNE     5$              ;BR IF YES
2750  010712  023737  002562  003504          CMP     DOCNT,MAXSK     ;DID ALL SEEKS?
2751  010720  001262                          BNE     1$              ;BR IF NO
2752
2753  010722  032737  001000  001246  5$:     BIT     #BIT9,$DEVM     ;DROP DX1?
2754  010730  001004                          BNE     6$              ;BR IF YES
2755  010732  023737  003032  003504          CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
2756  010740  001317                          BNE     3$              ;BR IF NO
2757
2758  010742  005737  003512          6$:     TST     COMP2           ;DOING COMPAT PASS 2?
2759  010746  001402                          BEQ     DXTST3          ;BR IF NO
2760  010750  000137  011706                  JMP     EOP             ;ELSE ALL DONE
2761
```

```
2762
2763                                    ;;*****************************************************************
2764                                    ;       INITIALIZE COMMAND TEST
2765                                    ;
2766                                    ;THE INITIALIZE  & EMPTY BUFFER COMMAND IS ISSUED TO DX0.
2767                                    ;DATA IS VERIFIED TO BE THAT OF TRACK 1, SECTOR 1.
2768                                    ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2769                                    ; 'INIT' CANNOT BE ISSUED TO DX1.
2770                                    ;;*****************************************************************
2771
2772  010754  032737  000400  001246  DXTST3: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2773  010762  001062                          BNE     DXTST4          ;BR IF YES
2774
2775  010764  012737  000003  003476          MOV     #3,DXTST        ;3 = INIT TEST
2776  010772  005037  002564                  CLR     DOERR           ;FLAGS
2777  010776  005037  002574                  CLR     DOCERR
2778
2779  011002  012737  016736  015226          MOV     #DOINIT,DODISP  ;SET DISPATCH TABLE TO GO TO INIT HANDLER
2780
2781  011010  005037  003500                  CLR     FIN
2782  011014  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2783
2784  011022  004737  013424                  JSR     PC,TIMDO
2785  011026  104046                          ERROR   46              ;DX0 HUNG ON INITIALIZE COMMAND
2786  011030  000437                          BR      DXTST4
2787
2788  011032  005737  177170                  TST     RXCS            ;ERROR?
2789  011036  100001                          BPL     1$              ;BR IF NO
2790  011040  104063                          ERROR   63              ;INIT COMMAND ERROR
2791
2792  011042  032737  000001  177172  1$:     BIT     #ID,RXES        ;INIT DONE BIT SET?
2793  011050  001001                          BNE     2$              ;BR IF YES
2794  011052  104055                          ERROR   55              ;INIT DONE NOT SET
2795
2796  011054  005037  002564          2$:     CLR     DOERR
2797  011060  005037  002574                  CLR     DOCERR
2798  011064  012737  000001  002566          MOV     #1,DOTRK        ;EXP TRK & SEC
2799  011072  012737  000001  002570          MOV     #1,DOSEC
2800
2801  011100  012737  016136  015226          MOV     #DOEBUF,DODISP  ;SETUP DISPATCH TABLE TO
2802                                                                  ;EMPTY BUFFER & VERIFY DATA
2803  011106  005037  003500                  CLR     FIN
2804  011112  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2805
2806  011120  004737  013424                  JSR     PC,TIMDO
2807  011124  104056                          ERROR   56              ;DX0 HUNG ON EMPTY BUFF COMMAND
2808  011126  000240                          NOP
```

```
2809
2810                                    ;;*****************************************************************
2811                                    ;       RESTORE COMMAND TEST
2812                                    ;
2813                                    ;THIS TEST VERIFIES THAT THE RESTORE COMMAND CAN BE ISSUED
2814                                    ;TO DX0 & DX1 & THAT NO ERRORS RESULT FROM IT.
2815                                    ;;*****************************************************************
2816
2817  011130  032737  000400  001246  DXTST4: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2818  011136  001050                          BNE     2$              ;BR IF YES
2819
2820  011140  012737  000004  003476          MOV     #4,DXTST        ;4 = RESTORE TEST
2821  011146  012737  016762  015226          MOV     #DOREST,DODISP  ;SETUP DISPATCH TABLE
2822  011154  005037  003500                  CLR     FIN
2823  011160  052737  000100  177170          BIS     #IE,RXCS
2824
2825  011166  004737  013424                  JSR     PC,TIMDO
2826  011172  104047                          ERROR   47              ;DX0 HUNG ON RESTORE COMMAND
2827  011174  000431                          BR      2$
2828
2829  011176  005737  177170                  TST     RXCS            ;ERROR?
2830  011202  100001                          BPL     1$              ;BR IF NO
2831  011204  104054                          ERROR   54              ;DX0 RESTORE COMMAND ERROR
2832
2833  011206  005037  002566          1$:     CLR     DOTRK           ;CHK FOR RECORD NOT FOUND (RNF)
2834  011212  012737  000001  002570          MOV     #1,DOSEC        ;SET TRK/SEC. SHOULD NOT MOVE
2835                                                                  ;LOGIC THINKS ITS ALREADY THERE.
2836  011220  005037  002564                  CLR     DOERR
2837  011224  005037  002574                  CLR     DOCERR
2838  011230  005037  003500                  CLR     FIN
2839  011234  012737  015632  015226          MOV     #DORSEC,DODISP  ;SETUP TO READ SECTOR
2840  011242  052737  000100  177170          BIS     #IE,RXCS        ;LET INTERR LOOSE
2841
2842  011250  004737  013424                  JSR     PC,TIMDO
2843  011254  104121                          ERROR   121             ;DX0 HUNG ON READ CMD
2844  011256  000240                          NOP
2845
2846  011260  032737  001000  001246  2$:     BIT     #BIT9,$DEVM     ;DROP DX1 TESTS?
2847  011266  001050                          BNE     DXTST5          ;BR IF YES
2848
2849  011270  012737  000004  003476          MOV     #4,DXTST        ;4 = RESTORE TEST
2850  011276  012737  017000  015234          MOV     #D1REST,D1DISP  ;REPEAT FOR DX1
2851  011304  005037  003500                  CLR     FIN
2852  011310  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE
2853
2854  011316  004737  013524                  JSR     PC,TIMD1
2855  011322  104122                          ERROR   122             ;DX1 HUNG ON RESTORE CMD
2856  011324  000431                          BR      DXTST5
2857
2858  011326  005737  177170                  TST     RXCS            ;ERROR?
2859  011332  100001                          BPL     3$              ;BR IF NO
2860  011334  104074                          ERROR   74              ;DX1 RESTORE CMD ERROR
```

```
2861
2862  011336  005037  003036          3$:     CLR     D1TRK           ;READ TO CHECK FOR RNF
2863  011342  012737  000001  003040          MOV     #1,D1SEC
2864  011350  005037  003034                  CLR     D1ERR
2865  011354  005037  003044                  CLR     D1CERR
2866  011360  005037  003500                  CLR     FIN
2867  011364  012737  017470  015234          MOV     #D1RSEC,D1DISP  ;SETUP TO READ SECTOR
2868  011372  052737  000120  177170          BIS     #<IE!DX1>,RXCS  ;LET INTERR LOOSE
2869
2870  011400  004737  013524                  JSR     PC,TIMD1
2871  011404  104123                          ERROR   123             ;DX1 HUNG ON READ CMD
2872  011406  000240                          NOP
2873
2874
2875
2876                                  ;;*****************************************************************
2877                                  ;       WRITE SECTOR WITH DELETED DATA MARK TEST
2878                                  ;
2879                                  ;THIS TEST VERIFIES THAT THE WRITE DELETED DATA COMMAND CAN BE ISSUED
2880                                  ;& THAT THE 'DELETED DATA' BIT 5 IS SET IN RXCS AFTER READY IS RECV'D.
2881                                  ;HARD ERRORS ARE THOSE ERRORS AFTER 10 MORE RETRIES.
2882                                  ;;*****************************************************************
2883
2884  011410  032737  000400  001246  DXTST5: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2885  011416  001043                          BNE     DXTST6          ;BR IF YES
2886
2887  011420  012737  000005  003476          MOV     #5,DXTST        ;5 = WR DEL DATA TEST
2888  011426  005037  002564                  CLR     D0ERR
2889  011432  005037  002574                  CLR     D0CERR
2890
2891  011436  012737  015244  015226          MOV     #D0FBUF,D0DISP  ;SETUP DISPATCH TABLE
2892
2893  011444  005037  003500                  CLR     FIN
2894  011450  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2895
2896  011456  004737  013424                  JSR     PC,TIMD0
2897  011462  104050                          ERROR   50              ;DX0 HUNG ON WRITE DEL DATA CMD
2898  011464  000420                          BR      DXTST6
2899
2900  011466  012737  017016  015226          MOV     #D0STAT,D0DISP  ;SETUP DISPATCH TABLE
2901
2902  011474  005037  003500                  CLR     FIN
2903  011500  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2904
2905  011506  004737  013424                  JSR     PC,TIMD0
2906  011512  104051                          ERROR   51              ;DX0 HUNG ON READ STATUS
2907  011514  000404                          BR      DXTST6
2908
2909  011516  005737  177170                  TST     RXCS            ;ERROR?
2910  011522  100001                          BPL     .+4             ;BR IF NO
2911  011524  104057                          ERROR   57              ;READ STATUS ERROR
```

```
2912
2913                                    ;*****************************************************************
2914                                    ;       INVALID ADDRESS TESTS
2915                                    ;
2916                                    ;A WRITE SECTOR COMMAND IS ISSUED TO INVALID TRACK 115(8)
2917                                    ;& RXCS IS CHECKED FOR THE INVALID ADDR BIT 1 TO BE SET AFTER RDY.
2918                                    ;THE ABOVE IS REPEATED FOR INVALID SECTOR 33(8).
2919                                    ;;*****************************************************************
2920
2921  011526  032737  000400  001246  DXTST6: BIT     #BIT8,$DEVM     ;DROP DX0 TESTS?
2922  011534  001062                          BNE     4$              ;BR IF YES
2923
2924  011536  005037  011704                  CLR     INVCT
2925  011542  012737  000006  003476          MOV     #6,DXTST        ;6 = INV ADDR TESTS
      011550  012737  000115  002566          MOV     #115,DOTRK
2927  011556  012737  000001  002570          MOV     #1,DOSEC
2928  011564  012737  046401  177174          MOV     #46401,RXSA     ;INVALID TRACK
2929
2930  011572  012737  017034  015226  1$:     MOV     #DOINV,DODISP   ;SETUP DISPATCH TABLE
2931
2932  011600  005037  003500                  CLR     FIN
2933  011604  052737  000100  177170          BIS     #IE,RXCS        ;LET LOOSE
2934
2935  011612  004737  013424                  JSR     PC,TIMDO
2936  011616  104060                          ERROR   60              ;DX0 HUNG ON INVALID ADDR
2937  011620  000432                          BR      EOP
2938
2939  011622  032737  000002  177172          BIT     #INVADR,RXES    ;INV ADDR BIT SET?
2940  011630  001002                          BNE     2$              ;BR IF YES
2941  011632  104061                          ERROR   61              ;INVALID ADDR BIT NOT SET
2942  011634  000404                          BR      3$
2943
2944  011636  005737  177170          2$:     TST     RXCS            ;ERROR BIT SET?
2945  011642  100401                          BMI     3$              ;BR IF YES
2946  011644  104062                          ERROR   62              ;ERROR BIT NOT SET
2947
2948  011646  005737  011704          3$:     TST     INVCT           ;DID WE JUST DO INV SECTOR?
2949  011652  001013                          BNE     4$              ;BR IF YES...DONE
2950
2951  011654  005237  011704                  INC     INVCT
2952  011660  005037  002566                  CLR     DOTRK
2953  011664  012737  000033  002570          MOV     #33,DOSEC
2954  011672  012737  000033  177174          MOV     #33,RXSA        ;INVALID SECTOR
2955  011700  000734                          BR      1$              ;REPEAT FOR INVALID SECTOR
2956
2957  011702  000401                  4$:     BR      .+4
2958
2959  011704  000000                  INVCT:  0                       ;0 = DOING INVALID TRACK
2960                                                                  ;1 - DOING INVALID SECTOR
2961
```

```
2962                                ;;*****************************************************************
2963                                ;       END OF PASS
2964                                ;;*****************************************************************
2965
2966  011706  005237  001176       EOP:    INC     $PASS           ;PASS CTR
2967  011712  005037  001200               CLR     $DEVCT          ;CLEAR SO DONT GET NEG NOS OR APT WILL BARF
2968  011716  004737  012624               JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
2969  011722  000437                         BR  3$              ;RET HERE IF NOT
2970  011724  104401  022633               TYPE    ,ENDPAS
2971  011730  013746  001176               MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
2972  011734  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2973
2974  011736  104401  022650               TYPE    ,MSG1
2975  011742  013746  031432               MOV     TERR,-(SP)      ;;SAVE TERR FOR TYPEOUT
2976  011746  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2977
2978  011750  104401  022677               TYPE    ,MSG2
2979  011754  013746  031434               MOV     TSERR,-(SP)     ;;SAVE TSERR FOR TYPEOUT
2980  011760  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2981
2982  011762  104401  022614               TYPE    ,SERIAL         ;SHOW SERIAL #
2983  011766  104401  030464               TYPE    ,$TTYIN
2984
2985  011772  104401  022733               TYPE    ,MSG3
2986  011776  013746  031436               MOV     PERR,-(SP)      ;;SAVE PERR FOR TYPEOUT
2987  012002  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2988
2989  012004  104401  022767               TYPE    ,MSG4
2990  012010  013746  031440               MOV     PSERR,-(SP)     ;;SAVE PSERR FOR TYPEOUT
2991  012014  104405                       TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
2992  012016  104401  012102               TYPE    ,NULL           ;NULL CHAR
2993  012022  012737  000114  003502  3$:  MOV     #114,MAXTRK     ;FOR ALL SUBSEQUENT PASSES
2994  012030  005737  003510               TST     COMP1
2995  012034  001006                       BNE     2$
2996  012036  005737  003512               TST     COMP2
2997  012042  001003                       BNE     2$
2998  012044  012737  000764  003504       MOV     #500.,MAXSK     ;SET ONLY IF NOT COMPATABLILTY TESTS
2999
3000  012052  005737  003510       2$:     TST     COMP1           ;COMPAT PASS 1?
3001  012056  001407                       BEQ     1$              ;BR IF NO
3002  012060  104401  023216               TYPE    ,COMPAT         ;DONE MSG
3003  012064  000000                       HALT
3004  012066  005037  003510               CLR     COMP1
3005  012072  005237  003512               INC     COMP2
3006  012076  000137  004364       1$:     JMP     LOOP            ;REPEAT
3007
3008  012102     377     377     000  NULL:  .BYTE   -1,-1,0        ;NULL CHAR STRING
3009          012106                       .EVEN
3010
```

```
3011
3012                                    ;;*****************************************************************
3013                                    ;SYSTEM SIZER SUBROUTINE
3014                                    ;;*****************************************************************
3015
3016  012106  012737  014174  000004    SIZE:   MOV     #INTSRV,ERRVEC  ;SETUP TIMEOUT TRAP ADDR
3017  012114  012737  000200  000006            MOV     #200,ERRVEC+2   ;LOCKOUT INTERR
3018  012122  012737  014174  000010            MOV     #INTSRV,RESVEC  ;SETUP RESERVED INST TRAP ADDR
3019  012130  012737  000200  000012            MOV     #200,RESVEC+2   ;LOCKOUT INTERR
3020
3021                                    ;;*****************************************************************
3022                                    ;ROUTINE TO SIZE THE AMOUNT OF MEMORY
3023                                    ;;*****************************************************************
3024  012136  005037  014202                    CLR     INTFLG
3025  012142  012700  003776                    MOV     #3776,R0        ;SIZE MEM. START WITH 1K
3026  012146  005001                            CLR     R1              ;CTR, 1=1K, 2=2K, ETC
3027  012150  005710                    1$:     TST     (R0)
3028  012152  005737  014202                    TST     INTFLG          ;GOT TIMEOUT INTERR?
3029  012156  001007                            BNE     2$              ;BR IF YES
3030  012160  005201                            INC     R1
3031  012162  020027  167776                    CMP     R0,#167776      ;DONE 32K MEM?
3032  012166  001405                            BEQ     3$              ;BR IF YES, ALL DONE
3033  012170  062700  004000                    ADD     #4000,R0        ;GO TO NEXT 1K
3034  012174  000765                            BR      1$
3035
3036  012176  162700  004000            2$:     SUB     #4000,R0        ;GO BACK TO LAST VALID 1K
3037  012202  010037  012622            3$:     MOV     R0,MAXMEM       ;SAVE
3038  012206  010146                            MOV     R1,-(SP)        ;;SAVE R1 FOR TYPEOUT
3039                                                                    ;;TYPE MEM SIZE
3040  012210  104405                            TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
3041  012212  104401  021725                    TYPE    ,MEM
3042
3043                                    ;;*****************************************************************
3044                                    ;ROUTINE TO CHECK IF "EIS-FIS" OPTION IS PRESENT
3045                                    ;;*****************************************************************
3046  012216  005037  012612                    CLR     EIPRES          ;CLEAR EIS-FIS PRESET FLAG
3047  012222  005037  014202                    CLR     INTFLG          ;CLEAR TRAP FLAG
3048  012226  072100                            ASH     R0,R1           ;EXECUTE EIS INSTRUCTION
3049  012230  000240                            NOP
3050  012232  005737  014202                    TST     INTFLG          ;TEST IF TRAP OCCURRED
3051  012236  001003                            BNE     60$             ;BR IF YES
3052  012240  005237  012612                    INC     EIPRES          ;NO TRAP - SET EIS-FIS PRESENT FLAG
3053  012244  000404                            BR      61$             ;BR OVER TYPE-OUT
3054  012246  104401  021705            60$:    TYPE    ,EISTXT         ;EIS-FIS TEXT
3055  012252  104401  022133                    TYPE    ,NOTPRES
3056  012256                            61$:
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO CHECK IF CLUSTER PORT IS PRESENT
                                ;;*****************************************************************

012256  005000                          CLR     R0
012260  005037  014202                  CLR     INTFLG
012264  012700  176500                  MOV     #TK1S,R0        ;SETUP CLUSTER CSR ADDR
012270  005710                  4$:     TST     (R0)            ;DARE IT TO TIMEOUT
012272  005737  014202                  TST     INTFLG          ;DID IT?
012276  001011                          BNE     6$              ;BR IF YES
012300  020027  176520                  CMP     R0,#TK3S        ;DID ALL 3?
012304  001403                          BEQ     5$              ;BR IF YES
012306  062700  000010                  ADD     #10,R0          ;ELSE DO NEXT
012312  000766                          BR      4$

012314  005237  012614          5$:     INC     CLPRES          ;OPTION PRESENT
012320  000406                          BR      7$              ;ALL 3 MUST RESPOND

012322  005037  012614          6$:     CLR     CLPRES          ;OPTION NOT PRESENT
012326  104401  021442                  TYPE    ,CLOPT
012332  104401  022133                  TYPE    ,NOTPRES

012336  012737  000006  000004  7$:     MOV     #ERRVEC+2,ERRVEC        ;RESET TIME-OUT VECTOR
012344  005037  000006                  CLR     ERRVEC+2
012350  012737  000012  000010          MOV     #RESVEC+2,RESVEC        ;RESET RESERVED INST VECTOR
012356  005037  000012                  CLR     RESVEC+2
```

```
3084
3085                                    ;;*****************************************************************
3086                                    ;ROUTINE TO TYPEOUT DROPPED ITEMS
3087                                    ;;*****************************************************************
3088  012362  005000                            CLR     R0
3089  012364  012701  000040                   MOV     #BIT5,R1
3090
3091  012370  030137  001246            8$:     BIT     R1,$DEVM        ;DEVICE DROPPED?
3092  012374  001006                            BNE     10$             ;BR IF YES
3093  012376  000241                    9$:     CLC
3094  012400  006301                            ASL     R1
3095  012402  103426                            BCS     13$             ;BR IF ALL BITS TESTED
3096  012404  062700  000002                   ADD     #2,R0           ;ELSE BUMP INDEX
3097  012410  000767                            BR      8$
3098
3099  012412  016037  012432  012422  10$:     MOV     12$(R0),11$
3100  012420  104401                            TYPE
3101  012422  000000                    11$:    .WORD   0               ;DEVICE TO BE DROPPED
3102  012424  104401  022446                   TYPE    ,DROP
3103  012430  000762                            BR      9$
3104
3105  012432  021604                    12$:    BUFTST
3106  012434  021705                            EISTXT
3107  012436  021676                            CLOCK
3108  012440  021572                            DRV0
3109  012442  021577                            DRV1
3110  012444  021515                            SCOM
3111  012446  021530                            ACOM
3112  012450  021462                            CLT1
3113  012452  021473                            CLT2
3114  012454  021504                            CLT3
3115  012456  021544                            PRINT
```

```
3116                                    ;;*****************************************************************
3117                                    ;ROUTINE TO CHECK IF PRINTER EXISTS
3118                                    ;;*****************************************************************
3119
3120  012460  005037  012616            13$:    CLR     PRPRES
3121  012464  005037  012620                    CLR     PRPORT
3122  012470  005737  177514                    TST     PRS             ;SEE IF PRINTER ERR
3123  012474  100035                            BPL     16$             ;BR IF NO
3124
3125  012476  032737  000010  001246            BIT     #BIT3,$DEVM     ;COMM PORT IN EXT LOOPBACK?
3126  012504  001405                            BEQ     15$             ;BR IF YES
3127
3128  012506  104401  021544            14$:    TYPE    ,PRINT
3129  012512  104401  022133                    TYPE    ,NOTPRES
3130  012516  000426                            BR      17$
3131
3132  012520  013737  003524  003534  15$:     MOV     AMBAUD,HOLD
3133  012526  052737  030034  003534            BIS     #<AMOD!OPAR!CHAR8>,HOLD
3134  012534  013737  003534  177420            MOV     HOLD,PARAM      ;SET PARAM REG
3135
3136  012542  042737  000006  176610            BIC     #<RTS!DTR>,AMRC ;MUST BE TOGGLED
3137  012550  052737  000006  176610            BIS     #<RTS!DTR>,AMRC
3138  012556  005737  177514                    TST     PRS             ;NOW ANY ERRORS?
3139  012562  100751                            BMI     14$             ;BR IF YES
3140  012564  005237  012620                    INC     PRPORT
3141  012570  005237  012616            16$:    INC     PRPRES          ;PRINTER PRESENT
3142
3143
3144
3145                                    ;;*****************************************************************
3146                                    ;SEE IF CONSOLE IS TO BE REMOVED
3147                                    ;;*****************************************************************
3148
3149  012574  032737  000020  001246  17$:     BIT     #BIT4,$DEVM
3150  012602  001402                            BEQ     18$             ;BR IF CONSOLE IS TO BE USED
3151  012604  104401  021631                    TYPE    ,NOCON          ;ELSE TELL OPERATOR
3152  012610  000207                    18$:    RTS     PC
3153
3154  012612  000000                    EIPRES: 0                       ;EIS-FIS PRESENT = NON-ZERO
3155  012614  000000                    CLPRES: 0                       ;CLUSTER PRESENT = NON-ZERO
3156  012616  000000                    PRPRES: 0                       ;PRINTER PRESENT = NON-ZERO
3157  012620  000000                    PRPORT: 0                       ;1 = PRINTER ENABLED THRU LOOPBACK
3158
3159  012622  000000                    MAXMEM: 0                       ;MAX MEMORY
3160
```

```
3161                                    ;;*****************************************************************
3162                                    ;ROUTINE TO SEE IF CONSOLE PRESENT
3163                                    ;RETURN IF NO
3164                                    ;RETURN + 2 IF YES
3165                                    ;;*****************************************************************
3166
3167  012624  004737  013620           CHKCON: JSR     PC,TIMER1       ;SPEND SOME TIME
3168  012630  105777  166314                   TSTB    @$TPS           ;XMIT RDY?
3169  012634  100002                           BPL     1$              ;BR IF NO  (NO CONSOLE)
3170  012636  062716  000002                   ADD     #2,(SP)         ;BUMP RETURN
3171  012642  000207                   1$:     RTS     PC
3172
3173
3174                                    ;;*****************************************************************
3175                                    ;ROUTINE TO FILL THE "WCP" TABLE WITH 125252 ON THE 1'ST PASS
3176                                    ;& RANDOM NUMBERS ON 2'ND & SUBSEQUENT PASSES.
3177                                    ;;*****************************************************************
3178
3179  012644  012700  000077           SETPAT: MOV     #77,R0          ;WORD COUNT
3180  012650  012702  003300                   MOV     #WCP,R2         ;TABLE ADDR
3181
3182  012654  005737  003510                   TST     COMP1           ;COMPATABLILTY TESTS DO WCP ONLY
3183  012660  001007                           BNE     5$
3184  012662  005737  003512                   TST     COMP2
3185  012666  001004                           BNE     5$
3186
3187  012670  032737  000001  001176           BIT     #BIT0,$PASS     ;CHECK ODD-EVEN PASS
3188  012676  001006                           BNE     2$              ;BR IF EVEN
3189
3190  012700  012701  125252           5$:     MOV     #125252,R1      ;ELSE LOAD WORST CASE PATT
3191  012704  010122                   1$:     MOV     R1,(R2)+
3192  012706  005300                           DEC     R0
3193  012710  001375                           BNE     1$
3194  012712  000414                           BR      4$
3195
3196  012714  005237  012752           2$:     INC     PATFLG
3197  012720  005037  013074                   CLR     LOLIM           ;SET LIMITS
3198  012724  012737  177777  013076           MOV     #-1,HILIM
3199  012732  004737  012754           3$:     JSR     PC,RAND         ;GET RANDOM NUMBER, RET WITH IT IN R1
3200  012736  010122                           MOV     R1,(R2)+
3201  012740  005300                           DEC     R0
3202  012742  001373                           BNE     3$
3203
3204  012744  005037  012752           4$:     CLR     PATFLG
3205  012750  000207                           RTS     PC
3206
3207  012752  000000                   PATFLG: .WORD 0                 ;FLAG USED BY "RAND" ROUTINE
```

```
3208
3209                                    ;;*****************************************************************
3210                                    ;       RANDOM NUMBER GENERATOR.  RETURN WITH NUMBER IN R1
3211                                    ;;*****************************************************************
3212
3213  012754  010046                   RAND:   MOV     R0,-(SP)        ;SAVE
3214  012756  010246                           MOV     R2,-(SP)
3215  012760  013700  013102           3$:     MOV     LONUM,R0
3216  012764  013701  013100                   MOV     HINUM,R1
3217  012770  012702  177771                   MOV     #-7,R2
3218  012774  006300                   1$:     ASL     R0
3219  012776  006101                           ROL     R1              ;ROTATE CARRY TO R1
3220  013000  005202                           INC     R2              ;DONE?
3221  013002  001374                           BNE     1$              ;BR IN NO
3222  013004  063700  013102                   ADD     LONUM,R0        ;ADD NUMBER TO MAKE X 129
3223  013010  005501                           ADC     R1
3224  013012  063701  013100                   ADD     HINUM,R1        ;ADD NUMBER TO MAKE X 129
3225  013016  062700  001057                   ADD     #1057,R0        ;ADD LO CONSTANT
3226  013022  005501                           ADC     R1
3227  013024  062701  047401                   ADD     #47401,R1       ;ADD HI CONSTANT
3228  013030  010037  013102                   MOV     R0,LONUM
3229  013034  010137  013100                   MOV     R1,HINUM
3230
3231  013040  005737  012752                   TST     PATFLG          ;FILLING DATA TABLE?
3232  013044  001002                           BNE     2$              ;BR IF YES
3233                                                                   ;ELSE SCALE TO BE WITHIN LIMITS
3234  013046  042701  177400                   BIC     #177400,R1      ;SAVE LO BYTE ONLY
3235  013052  020137  013074           2$:     CMP     R1,LOLIM
3236  013056  103740                           BLO     3$              ;BR IF N < LOLIM
3237  013060  020137  013076                   CMP     R1,HILIM
3238  013064  101335                           BHI     3$              ;BR IF N > HILIM
3239  013066  012602                           MOV     (SP)+,R2        ;RESTORE
3240  013070  012600                           MOV     (SP)+,R0
3241  013072  000207                           RTS     PC              ;ELSE EXIT WITH R1 = N
3242
3243  013074  000000                   LOLIM:  0
3244  013076  000000                   HILIM:  0
3245
3246  013100  176543                   HINUM:  .WORD 176543
3247  013102  123456                   LONUM:  .WORD 123456
3248
3249
3250  013104  123727  001210  000001  EXAMKB: CMPB    $ENV,#1         ;APT?
3251  013112  001404                           BEQ     1$              ;BR IF YES
3252  013114  005737  003514                   TST     SMFLG           ;SKIP IF DOING SYNC MODEM TEST
3253  013120  001001                           BNE     1$
3254  013122  104407                           CKSWR
3255  013124  000207                   1$:     RTS     PC
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO DISPLAY CURRENT DEVM & ALLOW OPERATOR TO INPUT NEW VALUE.
                                ;;*****************************************************************
013126  104401  023313    GT$DEV: TYPE    ,DEVM           ;ELSE SHOW CURRENT
013132  013746  001246            MOV     $DEVM,-(SP)     ;;SAVE $DEVM FOR TYPEOUT
013136  104402                    TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
013140  104401  030524            TYPE    ,$MNEW          ;GET NEW

013144  005046            1$:     CLR     -(SP)           ;CLR CTR
013146  005046                    CLR     -(SP)           ;CLR NEW DEVM

013150  105777  165770    2$:     TSTB    @$TKS           ;CHAR THERE?
013154  100375                    BPL     2$              ;BR IF NO
013156  117746  165764            MOVB    @$TKB,-(SP)     ;ELSE GET CHAR
013162  042716  177600            BIC     #^C177,(SP)     ;MAKE IT 7 BIT ASCII

013166  021627  000015    3$:     CMP     (SP),#15        ;<CR>?
013172  001013                    BNE     7$              ;BR IF NO
013174  005766  000004            TST     4(SP)           ;ELSE IS IT 1'ST CHAR?
013200  001403                    BEQ     4$              ;BR IF YES
013202  016637  000002  001246    MOV     2(SP),$DEVM     ;ELSE SAVE NEW DEVM
013210  062706  000006    4$:     ADD     #6,SP           ;RESTORE STACK
013214  104401  001165    5$:     TYPE    ,$CRLF
013220  000002            6$:     RTI

013222  004737  02, 24    7$:     JSR     PC,$TYPEC       ;ECHO CHAR
013226  021627  000060            CMP     (SP),#60        ;CHAR < 0?
013232  002420                    BLT     9$              ;BR IF YES
013234  021627  000067            CMP     (SP),#67        ;CHAR >7?
013240  003015                    BGT     9$              ;BR IF YES
013242  042726  000060            BIC     #60,(SP)+       ;STRIP OFF ASCII
013246  005766  000002            TST     2(SP)           ;IS IT 1'ST CHAR?
013252  001403                    BEQ     8$              ;BR IF YES
013254  006316                    ASL     (SP)            ;ELSE SHIFT PRESENT CHAR
013256  006316                    ASL     (SP)            ;TO MAKE ROOM
013260  006316                    ASL     (SP)            ;FOR NEW ONE

013262  005266  000002    8$:     INC     2(SP)           ;KEEP CHAR CT
013266  056616  177776            BIS     -2(SP),(SP)     ;SET IN NEW CHAR
013272  000726                    BR      2$              ;GET NEW ONE

013274  021627  000025    9$:     CMP     (SP),#25        ;CONTROL-U?
013300  001402                    BEQ     10$             ;BR IF YES
013302  104401  001164            TYPE    ,$QUES          ;ELSE TYPE ?<CRLF>
013306  062706  000006    10$:    ADD     #6,SP           ;RESTORE STACK
013312  000705                    BR      GT$DEV          ;TRY AGAIN
```

```
3305                            ;;*****************************************************************
3306                            ;* ROUTINE TO SET 2 CONSECUTIVE INTERRUPT VECTORS
3307                            ;* 3 ARGUMENTS ARE PASSED THRU R5:
3308                            ;*      VECTOR ADDRESS
3309                            ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR RECVR
3310                            ;*      INTERRUPT SERVICE ROUTINE ADDRESS FOR XMITTER.
3311                            ;* PRIORITY WILL BE SET TO DISABLE FURTHUR INTERR.
3312                            ;;*****************************************************************
3313
3314  013314  010046            SETVEC: MOV     R0,-(SP)                ;SAVE
3315  013316  012500                    MOV     (R5)+,R0                ;GET VECTOR ADDR
3316  013320  012520                    MOV     (R5)+,(R0)+             ;PUT SERV ROUTINE ADDR THERE
3317  013322  012720  000200            MOV     #200,(R0)+              ;DISABLE INTERR
3318  013326  012520                    MOV     (R5)+,(R0)+             ;GET XMIT SERV ADDR NEXT
3319  013330  012710  000200            MOV     #200,(R0)
3320  013334  012600                    MOV     (SP)+,R0                ;RESTORE
3321  013336  000205                    RTS     R5
3322
3323
3324
3325
3326
3327                            ;;*****************************************************************
3328                            ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
3329                            ;RETURN    IF TIMED OUT
3330                            ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3331                            ;;*****************************************************************
3332
3333  013340  012537  013420    TIMER:  MOV     (R5)+,FLGHLD    ;GET FLAG
3334  013344  012537  013422            MOV     (R5)+,CTHLD     ;GET COUNT THAT FLAG SHOULD GO TO
3335
3336  013350  012701  000006            MOV     #6,R1
3337  013354  012700  177777    4$:     MOV     #-1,R0
3338  013360  027737  000034  013422 1$: CMP     @FLGHLD,CTHLD   ;RESPONDING?
3339  013366  001411                    BEQ     2$              ;BR IF YES
3340  013370  004737  013104            JSR     PC,EXAMKB
3341  013374  005300                    DEC     R0              ;ELSE DEC COUNTER
3342  013376  001370                    BNE     1$              ;BR IF NOT TIMED OUT
3343  013400  005237  001200            INC     $DEVCT          ;FOR APT
3344  013404  005301                    DEC     R1
3345  013406  001362                    BNE     4$
3346  013410  000402                    BR      .+6             ;ELSE TIMED OUT
3347
3348  013412  062705  000004    2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
3349  013416  000205                    RTS     R5
3350
3351  013420  000000            FLGHLD: 0                       ;FLAG
3352  013422  000000            CTHLD:  0                       ;COUNT THAT FLAG MUST REACH
3353
```

```
3354                                    ;;*****************************************************************
3355                                    ;WATCHDOG TIMER TO PREVENT DX0 FROM ENTERING AN ENDLESS WAIT LOOP
3356                                    ;RETURN    IF TIMED OUT
3357                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3358                                    ;;*****************************************************************
3359
3360  013424  005037  013522           TIMD0:  CLR     D0TMO
3361  013430  012737  000002  013520           MOV     #2,CTREG1
3362  013436  012737  177777  013516   4$:     MOV     #-1,CTREG
3363  013444  005737  003500           1$:     TST     FIN             ;FINISHED?
3364  013450  001015                           BNE     2$              ;BR IF YES
3365  013452  004737  013104                   JSR     PC,EXAMKB
3366  013456  005337  013516                   DEC     CTREG           ;ELSE DEC CTR
3367  013462  001370                           BNE     1$              ;BR IF NOT TIMED OUT
3368  013464  005237  001200                   INC     $DEVCT          ;FOR APT
3369  013470  005337  013520                   DEC     CTREG1
3370  013474  001360                           BNE     4$
3371  013476  005237  013522                   INC     D0TMO           ;SET FLAG
3372  013502  000402                           BR      .+6             ;TIMED OUT
3373
3374  013504  005237  001200           2$:     INC     $DEVCT          ;FOR APT
3375  013510  062716  000004                   ADD     #4,(SP)         ;BUMP RET
3376  013514  000207                           RTS     PC
3377
3378  013516  000000                   CTREG:  0
3379  013520  000000                   CTREG1: 0
3380  013522  000000                   D0TMO:  0                       ;1 = TIMEOUT OCCURED
3381
3382
3383                                    ;;*****************************************************************
3384                                    ;WATCHDOG TIMER TO PREVENT DX1 FROM ENTERING AN ENDLESS WAIT LOOP
3385                                    ;RETURN    IF TIMED OUT
3386                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3387                                    ;;*****************************************************************
3388
3389  013524  005037  013616           TIMD1:  CLR     D1TMO
3390  013530  012737  000002  013520           MOV     #2,CTREG1
3391  013536  012737  177777  013516   4$:     MOV     #-1,CTREG
3392  013544  005737  003500           1$:     TST     FIN             ;FINISHED?
3393  013550  001015                           BNE     2$              ;BR IF YES
3394  013552  004737  013104                   JSR     PC,EXAMKB
3395  013556  005337  013516                   DEC     CTREG           ;ELSE DEC CTR
3396  013562  001370                           BNE     1$              ;BR IF NOT TIMED OUT
3397  013564  005237  001200                   INC     $DEVCT          ;FOR APT
3398  013570  005337  013520                   DEC     CTREG1
3399  013574  001360                           BNE     4$
3400  013576  005237  013616                   INC     D1TMO           ;SET FLAG
3401  013602  000402                           BR      .+6             ;TIMED OUT
3402
3403  013604  005237  001200           2$:     INC     $DEVCT          ;FOR APT
3404  013610  062716  000004                   ADD     #4,(SP)         ;BUMP RET
3405  013614  000207                           RTS     PC
3406
3407  013616  000000                   D1TMO:  0                       ;1 = TIMEOUT OCCURED
3408
```

```
3409                                    ;;*****************************************************************
3410                                    ;       GENERAL TIMER
3411                                    ;;*****************************************************************
3412
3413  013620  012700  177777            TIMER1: MOV     #-1,R0
3414  013624  005300                            DEC     R0
3415  013626  001376                            BNE     .-2
3416  013630  000207                            RTS     PC
3417
3418                                    ;;*****************************************************************
3419                                    ;WATCHDOG TIMER TO PREVENT DEVICES FROM ENTERING AN ENDLESS WAIT LOOP
3420                                    ;RETURN    IF TIMED OUT
3421                                    ;RETURN +4 IF NOT TIMED OUT & JUMP OVER ERROR
3422                                    ;;*****************************************************************
3423
3424  013632  012537  013712            TIMER2: MOV     (R5)+,REGHLD    ;GET REG
3425  013636  012537  013714                    MOV     (R5)+,BITHLD    ;GET BIT TO BE TESTED
3426
3427  013642  012701  000001                    MOV     #1,R1
3428  013646  012700  177777            4$:     MOV     #-1,R0
3429  013652  033777  013714  000032    1$:     BIT     BITHLD,@REGHLD  ;BIT THERE?
3430  013660  001011                            BNE     2$              ;BR IF YES
3431  013662  004737  013104                    JSR     PC,EXAMKB
3432  013666  005300                            DEC     R0              ;ELSE DEC COUNTER
3433  013670  001370                            BNE     1$              ;BR IF NOT TIMED OUT
3434  013672  005237  001200                    INC     $DEVCT          ;FOR APT
3435  013676  005301                            DEC     R1
3436  013700  001362                            BNE     4$
3437  013702  000402                            BR      .+6             ;ELSE TIMED OUT
3438
3439  013704  062705  000004            2$:     ADD     #4,R5           ;JUMP OVER ERROR ON RETURN
3440  013710  000205                            RTS     R5
3441
3442  013712  000000                    REGHLD: 0                       ;DEVICE REG
3443  013714  000000                    BITHLD: 0                       ;DEV REG BIT TO BE TESTED
3444
3445
3446                                    ;;*****************************************************************
3447                                    ;ROUTINE TO CLEAR BOTH BAD TRACK/SECTOR TABLES
3448                                    ;;*****************************************************************
3449
3450  013716  012700  003004            CLRBAD: MOV     #D0BAD,R0
3451  013722  005020                    1$:     CLR     (R0)+
3452  013724  020027  003030                    CMP     R0,#D0BAD+20.
3453  013730  001374                            BNE     1$
3454
3455  013732  012700  003254                    MOV     #D1BAD,R0
3456  013736  005020                    2$:     CLR     (R0)+
3457  013740  020027  003300                    CMP     R0,#D1BAD+20.
3458  013744  001374                            BNE     2$
3459
3460  013746  000207                            RTS     PC
3461
```

```
3462                                    ;;*****************************************************************
3463                                    ;LOAD DX0 BAD TRACK/SECTOR TABLE
3464                                    ;;*****************************************************************
3465
3466  013750  010046                    LDOBAD: MOV     R0,-(SP)        ;SAVE
3467
3468  013752  012700  003004                    MOV     #DOBAD,R0
3469  013756  005710                    1$:     TST     (R0)            ;ENTRY PRESENT?
3470  013760  001003                            BNE     2$              ;BR IF YES
3471  013762  013710  177174                    MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
3472  013766  000405                            BR      3$              ;EXIT
3473
3474  013770  005720                    2$:     TST     (R0)+           ;BUMP PTR
3475  013772  020027  003030                    CMP     R0,#DOBAD+20.   ;AT END OF TABLE?
3476  013776  001367                            BNE     1$              ;BR IF NO
3477  014000  104104                            ERROR   104             ;10 BAD SECTORS...REPLACE
3478  014002  012600                    3$:     MOV     (SP)+,R0        ;RESTORE
3479  014004  000207                            RTS     PC
3480
3481
3482
3483
3484
3485                                    ;;*****************************************************************
3486                                    ;CHECK DX0 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
3487                                    ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
3488                                    ;;*****************************************************************
3489
3490  014006  010046                    CKOBAD: MOV     R0,-(SP)        ;SAVE
3491  014010  010146                            MOV     R1,-(SP)
3492
3493  014012  013701  002566                    MOV     DOTRK,R1
3494  014016  000301                            SWAB    R1
3495  014020  053701  002570                    BIS     DOSEC,R1
3496
3497  014024  012700  003004                    MOV     #DOBAD,R0
3498  014030  020027  003030            1$:     CMP     R0,#DOBAD+20.   ;END OF TABLE?
3499  014034  001405                            BEQ     2$              ;BR IF YES
3500  014036  005710                            TST     (R0)            ;ELSE ANY ENTRY?
3501  014040  001403                            BEQ     2$              ;BR IF NO
3502  014042  022001                            CMP     (R0)+,R1        ;ELSE COMPARE?
3503  014044  001371                            BNE     1$              ;BR IF NO
3504  014046  000402                            BR      3$              ;ELSE DONT BUMP RET ADDR
3505
3506  014050  062705  000002            2$:     ADD     #2,R5           ;BUMP RET ADDR
3507  014054  012601                    3$:     MOV     (SP)+,R1        ;RESTORE
3508  014056  012600                            MOV     (SP)+,R0
3509  014060  000205                            RTS     R5
```

```
3510                                    ;;*****************************************************************
3511                                    ;LOAD DX1 BAD TRACK/SECTOR TABLE
3512                                    ;;*****************************************************************
3513
3514  014062  010046                    LD1BAD: MOV     R0,-(SP)        ;SAVE
3515
3516  014064  012700  003254                    MOV     #D1BAD,R0
3517  014070  005710                    1$:     TST     (R0)            ;ENTRY PRESENT?
3518  014072  001003                            BNE     2$              ;BR IF YES
3519  014074  013710  177174                    MOV     RXSA,(R0)       ;ELSE STORE BAD RXSA IN TABLE
3520  014100  000405                            BR      3$              ;EXIT
3521
3522  014102  005720                    2$:     TST     (R0)+           ;BUMP PTR
3523  014104  020027  003300                    CMP     R0,#D1BAD+20.   ;AT END OF TABLE?
3524  014110  001367                            BNE     1$              ;BR IF NO
3525  014112  104105                            ERROR   105             ;10 BAD SECTORS...REPLACE
3526  014114  012600                    3$:     MOV     (SP)+,R0        ;RESTORE
3527  014116  000207                            RTS     PC
3528
3529
3530
3531
3532
3533                                    ;;*****************************************************************
3534                                    ;CHECK DX1 BAD TRACK/SECTOR TABLE BEFORE DOING RANDOM SEEKS.
3535                                    ;IF TRK/SEC IN TABLE, RETURN TO RE-CALCULATE NEW TRACK & SECTOR.
3536                                    ;;*****************************************************************
3537
3538  014120  010046                    CK1BAD: MOV     R0,-(SP)        ;SAVE
3539  014122  010146                            MOV     R1,-(SP)
3540
3541  014124  013701  003036                    MOV     D1TRK,R1
3542  014130  000301                            SWAB    R1
3543  014132  053701  003040                    BIS     D1SEC,R1
3544
3545  014136  012700  003254                    MOV     #D1BAD,R0
3546  014142  020027  003300            1$:     CMP     R0,#D1BAD+20.   ;END OF TABLE?
3547  014146  001405                            BEQ     2$              ;BR IF YES
3548  014150  005710                            TST     (R0)            ;ELSE ANY ENTRY?
3549  014152  001403                            BEQ     2$              ;BR IF NO
3550  014154  022001                            CMP     (R0)+,R1        ;ELSE COMPARE?
3551  014156  001371                            BNE     1$              ;BR IF NO
3552  014160  000402                            BR      3$              ;ELSE DONT BUMP RET ADDR
3553
3554  014162  062705  000002            2$:     ADD     #2,R5           ;BUMP RET ADDR
3555  014166  012601                    3$:     MOV     (SP)+,R1        ;RESTORE
3556  014170  012600                            MOV     (SP)+,R0
3557  014172  000205                            RTS     R5
3558
```

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

014174  005237  014202          INTSRV: INC     INTFLG
014200  000002                          RTI
014202  000000                  INTFLG: 0


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

014204  013737  177514  014334  PRSRV:  MOV     PRS,SPRS                ;STORE
014212  005737  014334                  TST     SPRS            ;ERROR?
014216  100002                          BPL     1$              ;BR IF NO
014220  104007                          ERROR   7               ;PRINTER STATUS ERROR
014222  000002                          RTI

014224  013737  014330  177516  1$:     MOV     PRCHR,PRB       ;ELSE PRINT CHAR
014232  023727  014330  000015          CMP     PRCHR,#CR       ;JUST DID CR?
014240  001413                          BEQ     2$              ;BR IF YES
014242  023727  014330  000012          CMP     PRCHR,#LF       ;JUST DID LF?
014250  001413                          BEQ     3$              ;BR IF YES
014252  023727  014330  000176          CMP     PRCHR,#176      ;JUST DID LAST CHAR?
014260  001413                          BEQ     4$              ;BR IF YES
014262  005237  014330                  INC     PRCHR           ;ELSE BUMP CHAR FOR NEXT INTERRUPT
014266  000417                          BR      5$              ;EXIT

014270  012737  000012  014330  2$:     MOV     #LF,PRCHR       ;DO LF NEXT
014276  000413                          BR      5$
014300  012737  000040  014330  3$:     MOV     #40,PRCHR       ;DO SPACE NEXT
014306  000407                          BR      5$
014310  012737  000015  014330  4$:     MOV     #CR,PRCHR       ;DO CR NEXT & START OVER
014316  005237  014332                  INC     LINCT
014322  005237  001200                  INC     $DEVCT          ;FOR APT
014326  000002                  5$:     RTI

014330  000000                  PRCHR:  0                       ;CHAR TO BE PRINTED
014332  000000                  LINCT:  0                       ;LINE CTR
014334  000000                  SPRS:   0                       ;STORE PRINTER CSR
```

```
                                ;;*****************************************************************
                                ;CLUSTER TERMINAL #1 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
                                ;;*****************************************************************
014336  042737  000100  176504  TP1SRV: BIC     #IE,TP1S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
014344  013737  014432  176506          MOV     T1CHR,TP1B      ;PRINT CHAR
014352  000002                        RTI

014354  013737  176502  014434  TK1SRV: MOV     TK1B,T1HLD      ;STORE CHAR
014362  023737  014434  014432          CMP     T1HLD,T1CHR     ;OK?
014370  001401                        BEQ     1$              ;BR IF YES
014372  104010                        ERROR   10              ;CHAR COMP ERROR

014374  023727  014432  000377  1$:     CMP     T1CHR,#377      ;LAST CHAR?
014402  001403                        BEQ     2$              ;BR IF YES
014404  005237  014432                INC     T1CHR           ;ELSE BUMP
014410  000404                        BR      3$

014412  005037  014432          2$:     CLR     T1CHR           ;START OVER
014416  005237  001200                INC     $DEVCT          ;FOR APT
014422  052737  000100  176504  3$:     BIS     #IE,TP1S        ;ALLOW PRINTER INTERR
014430  000002                        RTI

014432  000000                  T1CHR:  0                       ;CHAR TO XMITT
014434  000000                  T1HLD:  0                       ;REC'D CHAR


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

014436  042737  000100  176514  TP2SRV: BIC     #IE,TP2S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
014444  013737  014532  176516          MOV     T2CHR,TP2B      ;PRINT CHAR
014452  000002                        RTI

014454  013737  176512  014534  TK2SRV: MOV     TK2B,T2HLD      ;STORE CHAR
014462  023737  014534  014532          CMP     T2HLD,T2CHR     ;OK?
014470  001401                        BEQ     1$              ;BR IF YES
014472  104011                        ERROR   11              ;CHAR COMP ERROR

014474  023727  014532  000377  1$:     CMP     T2CHR,#377      ;LAST CHAR?
014502  001403                        BEQ     2$              ;BR IF YES
014504  005237  014532                INC     T2CHR           ;ELSE BUMP
014510  000404                        BR      3$

014512  005037  014532          2$:     CLR     T2CHR           ;START OVER
014516  005237  001200                INC     $DEVCT          ;FOR APT
014522  052737  000100  176514  3$:     BIS     #IE,TP2S        ;ALLOW PRINTER INTERR
014530  000002                        RTI

014532  000000                  T2CHR:  0                       ;CHAR TO XMITT
014534  000000                  T2HLD:  0                       ;REC'D CHAR
```

```
                                ;;*****************************************************************
                                ;CLUSTER TERMINAL #3 INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
                                ;;*****************************************************************

014536  042737  000100  176524  TP3SRV: BIC     #IE,TP3S        ;DISABLE INTER, RECV'R WILL TURN BACK ON
014544  013737  014632  176526          MOV     T3CHR,TP3B      ;PRINT CHAR
014552  000002                  RTI

014554  013737  176522  014634  TK3SRV: MOV     TK3B,T3HLD      ;STORE CHAR
014562  023737  014634  014632          CMP     T3HLD,T3CHR     ;OK?
014570  001401                  BEQ     1$              ;BR IF YES
014572  104012                  ERROR   12              ;CHAR COMP ERROR

014574  023727  014632  000377  1$:     CMP     T3CHR,#377      ;LAST CHAR?
014602  001403                  BEQ     2$              ;BR IF YES
014604  005237  014632                  INC     T3CHR           ;ELSE BUMP
014610  000404                  BR      3$

014612  005037  014632          2$:     CLR     T3CHR           ;START OVER
014616  005237  001200                  INC     $DEVCT          ;FOR APT
014622  052737  000100  176524  3$:     BIS     #IE,TP3S        ;ALLOW PRINTER INTERR
014630  000002                  RTI

014632  000000                  T3CHR:  0                       ;CHAR TO XMITT
014634  000000                  T3HLD:  0                       ;REC'D CHAR

                                ;;*****************************************************************
                                ;ASYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 0 THRU 377.
                                ;;*****************************************************************

014636  042737  000100  176614  AMXSRV: BIC     #IE,AMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
014644  013737  014732  176616          MOV     COMCHR,AMXB     ;XMITT CHAR
014652  000002                  RTI

014654  013737  176612  014734  AMRSRV: MOV     AMRB,COMHLD     ;STORE CHAR
014662  023737  014734  014732          CMP     COMHLD,COMCHR   ;OK?
014670  001401                  BEQ     1$              ;BR IF YES
014672  104013                  ERROR   13              ;CHAR COMPARE ERROR

014674  023727  014732  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
014702  001403                  BEQ     2$              ;BR IF YES
014704  005237  014732                  INC     COMCHR          ;ELSE BUMP
014710  000404                  BR      3$

014712  005037  014732          2$:     CLR     COMCHR          ;START OVER
014716  005237  001200                  INC     $DEVCT          ;FOR APT
014722  052737  000100  176614  3$:     BIS     #IE,AMXC        ;ALLOW XMIT INTERR
014730  000002                  RTI

014732  000000                  COMCHR: 0                       ;XMIT CHAR
014734  000000                  COMHLD: 0                       ;REC'D CHAR
```

```
3708
3709                                    ;;*****************************************************************
3710                                    ;SYNC COMM PORT INTERRUPT HANDLER: LOOP VALUES 27 THRU 377.
3711                                    ;;*****************************************************************
3712
3713  014736  042737  000100  176624  SMXSRV: BIC     #IE,SMXC        ;DISABLE INTER, RECV'R WILL TURN BACK ON
3714  014744  013737  014732  176626          MOV     COMCHR,SMXB     ;XMITT CHAR
3715  014752  023727  014732  000377          CMP     COMCHR,#377     ;LAST CHAR?
3716  014760  001020                          BNE     2$              ;BR IF NO
3717
3718  014762  005737  015024                  TST     LSTCHR
3719  014766  001010                          BNE     1$
3720  014770  005237  015024                  INC     LSTCHR
3721  014774  052737  000100  176624          BIS     #IE,SMXC        ;ALLOW RDY INTR SO CAN SHUT DOWN FAST
3722  015002  005237  001200                  INC     $DEVCT          ;FOR APT
3723  015006  000002                          RTI
3724
3725  015010  042737  000020  176624  1$:     BIC     #SEND,SMXC      ;NO MORE TO SEND
3726  015016  005237  001200                  INC     $DEVCT          ;FOR APT
3727  015022  000002                  2$:     RTI
3728
3729  015024  000000                  LST CHR: 0                      ;SET WHEN LAST CHAR XMIT
3730
3731
3732
3733
3734
3735  015026  013737  176622  014734  SMRSRV: MOV     SMRB,COMHLD     ;STORE CHAR
3736  015034  123727  014734  000026          CMPB    COMHLD,#026     ;IGNORE SYNC CHARS
3737  015042  001430                          BEQ     4$
3738  015044  023737  014734  014732          CMP     COMHLD,COMCHR   ;OK?
3739  015052  001401                          BEQ     1$              ;BR IF YES
3740  015054  104014                          ERROR   14              ;CHAR COMPARE ERROR
3741
3742  015056  023727  014732  000377  1$:     CMP     COMCHR,#377     ;LAST CHAR?
3743  015064  001403                          BEQ     2$              ;BR IF YES
3744  015066  005237  014732                  INC     COMCHR          ;ELSE BUMP
3745  015072  000411                          BR      3$
3746
3747  015074  012737  177777  014732  2$:     MOV     #-1,COMCHR      ;INDICATE DONE
3748  015102  042737  000100  176620          BIC     #IE,SMRC        ;ALL DONE
3749  015110  005237  001200                  INC     $DEVCT          ;FOR APT
3750  015114  000403                          BR      4$
3751
3752  015116  052737  000100  176624  3$:     BIS     #IE,SMXC        ;ALLOW XMIT INTERR
3753  015124  000002                  4$:     RTI
3754
```

```
3755
3756                                    ;;*****************************************************************
3757                                    ;                   DISK INTERRUPT HANDLERS
3758                                    ;ALL DISK INTERRUPTS ENTER THRU RXSRV & DISPATCH TO THE
3759                                    ;CURRENT SERVICE ROUTINE POINTED TO BY:
3760                                    ;                   D0DISP FOR DX0 INTERRUPTS
3761                                    ;                   D1DISP FOR DX1 INTERRUPTS
3762                                    ;WILL GENERALLY BE IN THE ORDER OF SERVICE ROUTINES BELOW.
3763                                    ;;*****************************************************************
3764
3765  015126  105737  177170       RXSRV:  TSTB    RXCS            ;CONTR RDY?
3766  015132  100413                       BMI     1$              ;BR IF YES
3767  015134  013737  177170  015236       MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
3768  015142  013737  177172  015240       MOV     RXES,SRXES
3769  015150  013737  177174  015242       MOV     RXSA,SRXSA
3770  015156  104015                       ERROR   15              ;UNEXPECTED INTERRUPT
3771  015160  000002                       RTI                     ;GO BACK
3772
3773  015162  005737  177170       1$:     TST     RXCS            ;ERROR?
3774  015166  100011                       BPL     2$              ;BR IF NO
3775
3776  015170  013737  177170  015236       MOV     RXCS,SRXCS      ;ELSE SAVE FOR ERROR TYPEOUTS
3777  015176  013737  177172  015240       MOV     RXES,SRXES
3778  015204  013737  177174  015242       MOV     RXSA,SRXSA
3779
3780  015212  032737  000020  177170  2$:     BIT     #USEL,RXCS      ;CHECK UNIT SELECT BIT
3781  015220  001003                       BNE     D1INT           ;BR IF INTER FROM DX1
3782
3783  015222  000177  000000               JMP     @D0DISP         ;ELSE DISPATCH TO DX0 SERVICE ROUTINE
3784  015226  000000               D0DISP: 0
3785
3786  015230  000177  000000       D1INT:  JMP     @D1DISP         ;DX1 INTERRUPT DISPATCH
3787  015234  000000               D1DISP: 0
3788
3789
3790  015236  000000               SRXCS:  0                       ;SAVE RXCS
3791  015240  000000               SRXES:  0                       ;SAVE RXES
3792  015242  000000               SRXSA:  0                       ;SAVE RXSA
3793
```

```
3794
3795                                    ;;*****************************************************************
3796                                    ; HANDLER TO FILL THE DRIVE BUFFER FOR DX0
3797                                    ;& POINT TO THE WRITE SECTOR HANDLER.
3798                                    ;;*****************************************************************
3799
3800  015244  010246                    DOFBUF: MOV     R2,-(SP)        ;SAVE
3801  015246  012703  000100                    MOV     #100,R3         ;WORD CT
3802  015252  013701  002566                    MOV     DOTRK,R1
3803  015256  000301                            SWAB    R1
3804  015260  053701  002570                    BIS     DOSEC,R1
3805  015264  012737  015326  015226            MOV     #DOWSEC,DODISP  ;POINT TO WRITE SECTOR AFTER RTI
3806  015272  012737  000101  177170            MOV     #<FBUF!IE>,RXCS ;ISSUE FILL BUFFER CMD
3807  015300  010137  177172                    MOV     R1,RXDB         ;1'ST WORD = ADDRESS
3808  015304  005303                            DEC     R3
3809  015306  012702  003300                    MOV     #WCP,R2         ;GET TABLE ADDR
3810  015312  012237  177172            1$:     MOV     (R2)+,RXDB      ;ALL REST IS FROM TABLE
3811  015316  005303                            DEC     R3
3812  015320  001374                            BNE     1$
3813  015322  012602                            MOV     (SP)+,R2        ;RESTORE
3814  015324  000002                            RTI
3815
3816
3817
3818
3819
3820                                    ;;*****************************************************************
3821                                    ;HANDLER TO WRITE THE SECTOR FOR DX0 & POINT TO WRITE CHECK.
3822                                    ;;*****************************************************************
3823
3824  015326  013701  002566            DOWSEC: MOV     DOTRK,R1
3825  015332  000301                            SWAB    R1
3826  015334  053701  002570                    BIS     DOSEC,R1
3827  015340  010137  177174                    MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
3828  015344  012737  015402  015226            MOV     #DOWCHK,DODISP  ;POINT TO WRITE CHECK HANDLER
3829  015352  023727  003476  000005            CMP     DXTST,#5
3830  015360  001004                            BNE     1$
3831  015362  012737  000115  177170            MOV     #<WDDSEC!IE>,RXCS       ;ONLY FOR WRITE DELETED DATA TEST
3832  015370  000403                            BR      2$
3833  015372  012737  000105  177170    1$:     MOV     #<WSEC!IE>,RXCS ;ISSUE WRITE SECTOR COMMAND
3834  015400  000002                    2$:     RTI
3835
```

```
                        ;;*****************************************************************
                        ;HANDLER TO WRITE CHECK ON DX0.
                        ;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).
                        ;;*****************************************************************

015402  005737  177170  DOWCHK: TST     RXCS            ;ERROR?
015406  100033                  BPL     1$              ;BR IN NO
015410  023727  002564  000012        CMP     DOERR,#10.      ;10 ERRORS?
015416  001066                  BNE     6$              ;BR IF NO & TRY AGAIN
015420  004737  013750          JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
015424  032737  000020  015240        BIT     #RNF,SRXES      ;SEEK ERROR?
015432  001417                  BEQ     7$              ;BR IF NO
015434  004737  020612          JSR     R7,SEKTST       ;SEE IF REALLY ON RIGHT TRACK
015440  005737  003516          TST     SEKFLG
015444  001412                  BEQ     7$              ;BR IF YES
015446  005037  003516          CLR     SEKFLG          ;ELSE CLR FLG
015452  104064                  ERROR   64              ;HARD SEEK ERROR WRITE SECTOR
015454  012737  015534  015226        MOV     #2$,DODISP      ;GO TO 2$ AFTER RESTORE
015462  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
015470  000002                  RTI

015472  104016          7$:     ERROR   16              ;HARD ERROR WRITE SECTOR
015474  000417                  BR      2$              ;GO TO NEXT SECTOR

015476  005737  002564  1$:     TST     DOERR           ;ANY PREV ERRORS?
015502  001414                  BEQ     2$              ;BR IF NO
015504  005237  031434          INC     TSERR           ;TOTAL SOFT ERR CT
015510  005237  031440          INC     PSERR           ;PASS  SOFT ERR CT
015514  005737  003516          TST     SEKFLG          ;SEEK ERROR?
015520  001404                  BEQ     8$              ;BR IF NO
015522  005037  003516          CLR     SEKFLG
015526  104065                  ERROR   65              ;SOFT SEEK ERROR WRITE SECTOR
015530  000401                  BR      2$
015532  104017          8$:     ERROR   17              ;SOFT ERROR WRITE SECTOR

015534  005037  002564  2$:     CLR     DOERR
015540  004537  020532          JSR     R5,NXTSEC       ;CALC NEXT SECTOR
015544  002570                    DOSEC
015546  000137  015244  4$:     JMP     DOFBUF          ;GO TO FILL BUFFER & DO ANOTHER

015552  012737  000001  002570  5$:     MOV     #1,DOSEC        ;ALL  SECTORS DONE...CHECK DATA
015560  005037  002564          CLR     DOERR
015564  005037  002572          CLR     DOCMER
015570  000137  015632          JMP     DORSEC          ;GO TO READ SECTOR HANDLER

015574  005237  002564  6$:     INC     DOERR
015600  032737  000020  015240        BIT     #RNF,SRXES      ;SEEK ERROR?
015606  001757                  BEQ     4$              ;BR IN NO
015610  004737  020612          JSR     R7,SEKTST       ;SEE IF ON TRACK
015614  012737  015244  015226        MOV     #DOFBUF,DODISP  ;GOTO FILL BUFF AFTER RESTORE
015622  012737  000117  177170        MOV     #<RESTOR!IE>,RXCS
015630  000002                  RTI
```

```
3892
3893                            ;;*****************************************************************
3894                            ;HANDLER TO READ A SECTOR ON DX0 & POINT TO CHECK.
3895                            ;;*****************************************************************
3896
3897  015632  013701  002566    DORSEC: MOV     DOTRK,R1
3898  015636  000301                    SWAB    R1
3899  015640  053701  002570            BIS     DOSEC,R1
3900  015644  010137  177174            MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR
3901
3902  015650  012737  015666  015226      MOV     #DORCHK,DODISP  ;POINT TO READ CHECK HANDLER
3903  015656  012737  000107  177170      MOV     #<RSEC!IE>,RXCS ;ISSUE READ SECTOR COMMAND
3904  015664  000002                      RTI
3905
3906                            ;;*****************************************************************
3907                            ;HANDLER TO READ CHECK ON DX0.
3908                            ;WILL GO TO NEXT SECTOR/TRK IF HARD ERROR.
3909                            ;WILL GO TO EMPTY BUFFER IF NO ERROR OR SOFT ERROR.
3910                            ;WILL GO TO READ SECTOR (SAME) ON ERROR.
3911                            ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
3912                            ;;*****************************************************************
3913
3914  015666  005737  177170    DORCHK: TST     RXCS            ;ERROR?
3915  015672  100065                    BPL     1$              ;BR IF NO
3916  015674  023727  002564  000012      CMP     DOERR,#10.      ;10 ERRORS?
3917  015702  001417                      BEQ     3$              ;BR IF YES
3918  015704  005237  002564            INC     DOERR           ;ELSE DO AGAIN
3919  015710  032737  000020  015240      BIT     #RNF,SRXES      ;SEEK ERROR?
3920  015716  001745                      BEQ     DORSEC          ;BR IF NO
3921  015720  004737  020612            JSR     R7,SEKTST       ;SEE IF ON TRACK
3922  015724  012737  015632  015226      MOV     #DORSFC,DODISP  ;ELSE DO RESTORE & RET TO RD SEC
3923  015732  012737  000117  177170      MOV     #<RESTOR!IE>,RXCS
3924  015740  000002                      RTI
3925
3926  015742  004737  013750      3$:     JSR     PC,LDOBAD       ;LOAD BAD TRK/SEC TABLE
3927  015746  032737  000020  015240      BIT     #RNF,SRXES      ;SEEK ERROR?
3928  015754  001417                      BEQ     7$              ;BR IF NO
3929  015756  004737  020612            JSR     R7,SEKTST       ;SEE IF REALLY ON RIGHT TRACK
3930  015762  005737  003516            TST     SEKFLG
3931  015766  001422                      BEQ     10$             ;BR IF YES
3932  015770  005037  003516            CLR     SEKFLG          ;ELSE CLR FLG
3933  015774  104066                    ERROR   66              ;SEEK ERROR READ SECTOR
3934
3935  015776  012737  016714  015226      MOV     #DONEXT,DODISP  ;DO RESTORE
3936  016004  012737  000117  177170      MOV     #<RESTOR!IE>,RXCS
3937  016012  000002                      RTI
3938
3939  016014  032737  000010  015240  7$: BIT     #CRC,SRXES      ;CRC ERROR?
3940  016022  001404                      BEQ     10$             ;BR IF NO
3941  016024  005237  002602            INC     DOCRC           ;ELSE SET FLAG
3942  016030  104106                    ERROR   106             ;HARD CRC ERROR
3943  016032  000424                      BR      2$              ;& GO TO EMP BUFF TO CHK DATA
3944
3945  016034  005037  002602      10$:    CLR     DOCRC
3946  016040  104020                    ERROR   20              ;HARD READ ERROR
3947  016042  000137  016714            JMP     DONEXT
```

```
3948
3949  016046  005737  002564          1$:     TST     DOERR           ;ANY PREV ERRORS?
3950  016052  001414                          BEQ     2$              ;BR IF NO
3951  016054  005237  031434                  INC     TSERR           ;TOTAL SOFT ERR CT
3952  016060  005237  031440                  INC     PSERR           ;PASS  SOFT ERR CT
3953  016064  005737  003516                  TST     SEKFLG          ;SEEK ERROR?
3954  016070  001404                          BEQ     9$              ;BR IF NO
3955  016072  005037  003516                  CLR     SEKFLG
3956  016076  104067                          ERROR   67              ;SEEK ERROR READ SECTOR
3957  016100  000401                          BR      2$
3958  016102  104021                  9$:     ERROR   21              ;SOFT ERROR  READ SECTOR
3959
3960  016104  005037  002564          2$:     CLR     DOERR
3961  016110  023727  003476  000005          CMP     DXTST,#5        ;DOING WRITE DELETED DATA TESTS?
3962  016116  001005                          BNE     6$              ;BR IF NO
3963  016120  032737  000040  177172          BIT     #DD,RXES        ; 'DELETED DATA' SET?
3964  016126  001001                          BNE     6$              ;BR IF YES
3965  016130  104052                          ERROR   52              ;DD NOT SET
3966
3967  016132  000137  016136          6$:     JMP     DOEBUF          ;GO TO EMPTY BUFFER
3968
3969
3970
3971
3972
3973
3974                                  ;;*******************************************************************
3975                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX0
3976                                  ;& POINT TO THE CHECK DATA HANDLER.
3977                                  ;;*******************************************************************
3978
3979  016136  012703  000100          DOEBUF: MOV     #100,R3         ;WORD CT
3980  016142  012701  002604                  MOV     #DOBUF,R1       ;BUFFER ADDRESS
3981  016146  012737  016174  015226          MOV     #DOCDAT,DODISP  ;POINT TO CHK DATA AFT INTER.
3982  016154  012737  000103  177170          MOV     #<EBUF!IE>,RXCS ;ISSUE EMPTY BUFFER CMD
3983  016162  013721  177172          1$:     MOV     RXDB,(R1)+      ;GET WORD & STORE IT
3984  016166  005303                          DEC     R3
3985  016170  001374                          BNE     1$
3986  016172  000002                          RTI
3987
3988
```

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

016174  005037  002572  DOCDAT: CLR     DOCMER
016200  012703  000100           MOV     #100,R3          ;WORD CTR
016204  012704  003276           MOV     #WCP-2,R4        ;GET TABLE ADDR
016210  013701  002566           MOV     DOTRK,R1
016214  000301                   SWAB    R1
016216  053701  002570           BIS     DOSEC,R1         ;R1 NOW HAS EXPECTED DATA (1'ST WORD)
016222  012702  002604           MOV     #DOBUF,R2        ;GET BUFFER ADDR
016226  020112                   CMP     R1,(R2)          ;1'ST WORD COMPARE OK?
016230  001414                   BEQ     2$               ;BR IF YES
016232  010137  002576           MOV     R1,DOEXP         ;ELSE SAVE
016236  011237  002600  6$:      MOV     (R2),DOREC
016242  005237  002572           INC     DOCMER
016246  000414                   BR      3$

016250  021412           1$:     CMP     (R4),(R2)        ;COMPARE ALL REST
016252  001403                   BEQ     2$               ;BR IF OK
016254  011437  002576           MOV     (R4),DOEXP       ;ELSE SAVE
016260  000766                   BR      6$

016262  005303           2$:     DEC     R3
016264  001405                   BEQ     3$
016266  062702  000002           ADD     #2,R2
016272  062704  000002           ADD     #2,R4
016276  000764                   BR      1$

016300  005737  002572   3$:     TST     DOCMER           ;ANY COMP ERR?
016304  001421                   BEQ     4$               ;BR IF NO
016306  005737  002602           TST     DOCRC            ;HERE FROM CRC ERROR?
016312  001402                   BEQ     14$              ;BR IF NO
016314  104107                   ERROR   107              ;DATA CRC ERROR
016316  000431                   BR      5$

016320  023727  002574  000012  14$:     CMP     DOCERR,#10.      ;ELSE, 10 ERRS YET?
016326  001033                   BNE     7$               ;BR IF NO
016330  023727  003476  000003           CMP     DXTST,#3         ;DOING INIT TEST?
016336  001002                   BNE     8$               ;BR IN NO
016340  104053                   ERROR   53               ;INITIALIZE ERROR
016342  000417                   BR      5$

016344  104022           8$:     ERROR   22               ;HARD ERROR  DATA COMPARE
016346  000415                   BR      5$

016350  005737  002602   4$:     TST     DOCRC            ;HERE FROM CRC ERROR?
016354  001402                   BEQ     15$              ;BR IF NO
016356  104110                   ERROR   110              ;CRC ERR WITH DATA OK
016360  000410                   BR      5$
```

```

016362  005737  002574          15$:    TST     DOCERR          ;ANY PREV ERR?
016366  001405                          BEQ     5$              ;BR IF NO
016370  005237  031434                  INC     TSERR           ;TOTAL SOFT ERR CT
016374  005237  031440                  INC     PSERR           ;PASS  SOFT ERR CT
016400  104023                          ERROR   23              ;SOFT ERROR  DATA COMP
016402  005037  002574          5$:     CLR     DOCERR
016406  005037  002602                  CLR     DOCRC
016412  000137  016714                  JMP     DONEXT

016416  005237  002574          7$:     INC     DOCERR
016422  023727  003476  000003          CMP     DXTST,#3        ;DOING INIT TEST?
016430  001402                          BEQ     12$             ;BR IF YES
016432  000137  015632                  JMP     DORSEC          ;ELSE READ SECTOR OVER AGAIN
016436  000137  017052          12$:    JMP     DODUN           ;ALL DONE



                                ;;*************************************************************
                                ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX0.
                                ;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.
                                ;;*************************************************************

016442  005037  002564          DONXT1: CLR     DOERR
016446  005037  002572                  CLR     DOCMER
016452  004537  020532                  JSR     R5,NXTSEC       ;CALC NEXT SECTOR
016456  002570                            DOSEC
016460  000137  015632          2$:     JMP     DORSEC          ;GO READ SECTOR

016464  004737  012624          3$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
016470  000415                                  BR      6$      ;RET HERE IF NOT
016472  032777  010000  162440          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
016500  001411                          BEQ     6$
016502  104401  021572                  TYPE    ,DRV0
016506  104401  022504                  TYPE    ,TRK
016512  013746  002566                  MOV     DOTRK,-(SP)     ;;SAVE DOTRK FOR TYPEOUT
016516  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
016520  104401  022574                  TYPE    ,DUN
016524  005237  003500          6$:     INC     FIN             ;SETUP TO GO TO DX1
016530  023737  002566  003502          CMP     DOTRK,MAXTRK    ;LAST TRACK?
016536  001406                          BEQ     4$              ;BR IF YES
016540  005237  002566                  INC     DOTRK           ;BUMP TRACK
016544  012737  000001  002570          MOV     #1,DOSEC        ;INIT SECTOR
016552  000417                          BR      5$

016554  004737  012624          4$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
016560  000412                                  BR      7$      ;RET HERE IF NOT
016562  032777  010000  162350          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
016570  001406                          BEQ     7$
016572  104401  021572                  TYPE    ,DRV0
016576  104401  022511                  TYPE    ,PATT
016602  104401  022574                  TYPE    ,DUN
016606  005237  002560          7$:     INC     DOPAT           ;PATT DONE FLAG
016612  042737  000100  177170  5$:     BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
016620  000002                          RTI
```

```
                                ;;*****************************************************************
                                ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX0.
                                ;;*****************************************************************

016622  005037  002564          DONXT2: CLR     DOERR
016626  005037  002572                  CLR     DOCMER
016632  005237  003500                  INC     FIN             ;SETUP TO GO TO DX1
016636  005237  002562                  INC     DOCNT
016642  023737  002562  003504          CMP     DOCNT,MAXSK     ;DID ALL SEEKS?
016650  001015                          BNE     1$              ;BR IF NO
016652  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
016656  000412                              BR  1$              ;RET HERE IF NOT
016660  032777  010000  162252          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
016666  001406                          BEQ     1$
016670  104401  021572                  TYPE    ,DRV0
016674  104401  022524                  TYPE    ,RANDOM
016700  104401  022574                  TYPE    ,DUN
016704  042737  000100  177170  1$:     BIC     #IE,RXCS        ;DISABLE DX0 INTERRUPTS
016712  000002                          RTI


016714  023727  003476  000001  DONEXT: CMP     DXTST,#1
016722  001647                          BEQ     DONXT1
016724  023727  003476  000002          CMP     DXTST,#2
016732  001733                          BEQ     DONXT2
016734  000446                          BR      DODUN
```

```
4129
4130                                    ;;*****************************************************************
4131                                    ;HANDLER TO SETUP INITIALIZE COMMAND TEST
4132                                    ;;*****************************************************************
4133
4134  016736  012737  046032  177174  DOINIT: MOV     #46032,RXSA     ;TRY TO FAKE DRIVE TO GO TO TRK 76 SEC 26
4135                                                                    ;INIT SHOULD GO TO TRK 1, SEC 1.
4136  016744  012737  017052  015226           MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
4137  016752  012737  000111  177170           MOV     #<INITAL!IE>,RXCS       ;DO INIT COMMAND
4138  016760  000002                           RTI
4139
4140
4141                                    ;;*****************************************************************
4142                                    ;HANDLERS TO SETUP RESTORE COMMAND TEST
4143                                    ;;*****************************************************************
4144
4145  016762  012737  017052  015226  DOREST: MOV     #DODUN,DODISP   ;POINT TO DONE HANDLER
4146  016770  012737  000117  177170           MOV     #<RESTOR!IE>,RXCS
4147  016776  000002                           RTI
4148
4149  017000  012737  017066  015234  D1REST: MOV     #D1DUN,D1DISP
4150  017006  012737  000137  177170           MOV     #<RESTOR!IE!DX1>,RXCS
4151  017014  000002                           RTI
4152
4153                                    ;;*****************************************************************
4154                                    ;HANDLER TO SETUP READ STATUS COMMAND TEST
4155                                    ;;*****************************************************************
4156
4157  017016  012737  017052  015226  DOSTAT: MOV     #DODUN,DODISP   ;POINT TO DONE
4158  017024  012737  000113  177170           MOV     #<RSTAT!IE>,RXCS        ;DO READ STATUS COMMAND
4159  017032  000002                           RTI
4160
4161                                    ;;*****************************************************************
4162                                    ;HANDLER TO START INVALID ADDRESS TESTS
4163                                    ;;*****************************************************************
4164
4165  017034  012737  017052  015226  DOINV:  MOV     #DODUN,DODISP   ;POINT TO DONE
4166  017042  012737  000105  177170           MOV     #<WSEC!IE>,RXCS ;DO WRITE SECTOR COMMAND
4167  017050  000002                           RTI
4168
4169
4170                                    ;;*****************************************************************
4171                                    ;HANDLERS TO FINISH INITIALIZE, RESTORE, WRITE DELETED DATA,
4172                                    ;READ STATUS & INVALID ADDRESS TESTS.
4173                                    ;;*****************************************************************
4174
4175  017052  005237  003500           DODUN:  INC     FIN             ;SET DONE FLAG
4176  017056  042737  000100  177170           BIC     #IE,RXCS        ;DISABLE FURTHUR DX0 INTERRUPTS
4177  017064  000002                           RTI
4178
4179  017066  005237  003500           D1DUN:  INC     FIN
4180  017072  042737  000120  177170           BIC     #<IE'DX1>,RXCS
4181  017100  000002                           RTI
4182
```

```
4183
4184
4185                                    ;;*****************************************************************
4186                                    ; HANDLER TO FILL THE DRIVE BUFFER FOR DX1
4187                                    ;& POINT TO THE WRITE SECTOR HANDLER.
4188                                    ;;*****************************************************************
4189
4190  017102  010246                   D1FBUF: MOV     R2,-(SP)        ;SAVE
4191  017104  012703  000100                   MOV     #100,R3         ;WORD CT
4192  017110  013701  003036                   MOV     D1TRK,R1
4193  017114  000301                           SWAB    R1
4194  017116  053701  003040                   BIS     D1SEC,R1
4195
4196  017122  005737  003510                   TST     COMP1           ;DONT SET BIT15 IN COMPATABILTIY TESTS
4197  017126  001005                           BNE     2$
4198  017130  005737  003512                   TST     COMP2
4199  017134  001002                           BNE     2$
4200  017136  052701  100000                   BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA
4201
4202  017142  012737  017204  015234   2$:     MOV     #D1WSEC,D1DISP  ;POINT TO WRITE SECTOR AFTER RTI
4203  017150  012737  000121  177170           MOV     #<FBUF!IE!DX1>,RXCS     ;ISSUE FILL BUFFER CMD
4204  017156  010137  177172                   MOV     R1,RXDB         ;1'ST WORD = ADDRESS
4205  017162  005303                           DEC     R3
4206  017164  012702  003300                   MOV     #WCP,R2 ;GET TABLE ADDR
4207  017170  012237  177172           1$:     MOV     (R2)+,RXDB
4208  017174  005303                           DEC     R3
4209  017176  001374                           BNE     1$
4210  017200  012602                           MOV     (SP)+,R2        ;RESTORE
4211  017202  000002                           RTI
4212
4213
4214                                    ;;*****************************************************************
4215                                    ;HANDLER TO WRITE THE SECTOR FOR DX1 & POINT TO WRITE CHECK.
4216                                    ;;*****************************************************************
4217
4218  017204  013701  003036           D1WSEC: MOV     D1TRK,R1
4219  017210  000301                           SWAB    R1
4220  017212  053701  003040                   BIS     D1SEC,R1
4221  017216  010137  177174                   MOV     R1,RXSA ;LOAD TRACK & SECTOR ADDR
4222  017222  012737  017240  015234           MOV     #D1WCHK,D1DISP  ;POINT TO WRITE CHECK HANDLER
4223  017230  012737  000125  177170           MOV     #<WSEC!IE!DX1>,RXCS     ;ISSUE WRITE SECTOR COMMAND
4224  017236  000002                           RTI
4225
```

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

017240  005737  177170        D1WCHK: TST     RXCS            ;ERROR?
017244  100033                        BPL     1$              ;BR IN NO
017246  023727  003034  000012        CMP     D1ERR,#10.      ;10 ERRORS?
017254  001066                        BNE     6$              ;BR IF NO & TRY AGAIN
017256  004737  014062                JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
017262  032737  000020  015240        BIT     #RNF,SRXES      ;SEEK ERROR?
017270  001417                        BEQ     7$              ;BR IF NO
017272  004737  020612                JSR     R7,SEKTST       ;SEE IF REALLY ON RIGHT TRACK
017276  005737  003516                TST     SEKFLG
017302  001412                        BEQ     7$              ;BR IF YES
017304  005037  003516                CLR     SEKFLG          ;ELSE CLR FLG
017310  104070                        ERROR   70              ;SEEK ERROR WRITE SECTOR
017312  012737  017372  015234        MOV     #2$,D1DISP      ;RET TO 2$ AFT RESTORE
017320  012737  000137  177170        MOV     #<RESTOR!IE!DX1>,RXCS
017326  000002                        RTI

017330  104024                7$:     ERROR   24              ;ERROR WRITE SECTOR
017332  000417                        BR      2$              ;GO TO NEXT SECTOR

017334  005737  003034        1$:     TST     D1ERR           ;ANY PREV ERRORS?
017340  001414                        BEQ     2$              ;BR IF NO
017342  005237  031434                INC     TSERR           ;TOTAL SOFT ERR CT
017346  005237  031440                INC     PSERR           ;PASS  SOFT ERR CT
017352  005737  003516                TST     SEKFLG          ;SEEK ERROR?
017356  001404                        BEQ     8$              ;BR IF NO
017360  005037  003516                CLR     SEKFLG
017364  104071                        ERROR   71              ;SEEK ERROR WRITE SECTOR
017366  000401                        BR      2$
017370  104025                8$:     ERROR   25              ;ERROR WRITE SECTOR

017372  005037  003034        2$:     CLR     D1ERR
017376  004537  020532                JSR     R5,NXTSEC       ;CALC NEXT SECTOR
017402  003040                          D1SEC
017404  000137  017102        4$:     JMP     D1FBUF          ;GO TO FILL BUFFER & DO ANOTHER

017410  012737  000001  003040  5$:   MOV     #1,D1SEC        ;ALL SECTORS DONE...CHECK DATA
017416  005037  003034                CLR     D1ERR
017422  005037  003042                CLR     D1CMER
017426  000137  017470                JMP     D1RSEC          ;GO TO READ SECTOR HANDLER

017432  005237  003034        6$:     INC     D1ERR
017436  032737  000020  015240        BIT     #RNF,SRXES      ;SEEK ERROR?
017444  001757                        BEQ     4$              ;BR IF NO
017446  004737  020612                JSR     R7,SEKTST       ;SEE IF ON TRACK
017452  012737  017102  015234        MOV     #D1FBUF,D1DISP  ;RET TO FILL BUFF AFT RESTORE
017460  012737  000137  177170        MOV     #<RESTOR!IE!DX1>,RXCS
017466  000002                        RTI
```

```
4281
4282                                    ;;*****************************************************************
4283                                    ;HANDLER TO READ A SECTOR ON DX1 & POINT TO CHECK.
4284                                    ;;*****************************************************************
4285
4286  017470  013701  003036           D1RSEC: MOV     D1TRK,R1
4287  017474  000301                           SWAB    R1
4288  017476  053701  003040                   BIS     D1SEC,R1
4289  017502  010137  177174                   MOV     R1,RXSA ;LOAD TRK & SECTOR ADDR
4290
4291  01750o  012737  017524  015234           MOV     #D1RCHK,D1DISP  ;POINT TO READ CHECK HANDLER
4292  017514  012737  000127  177170           MOV     #<RSEC!IE!DX1>,RXCS     ;ISSUE READ SECTOR COMMAND
4293  017522  000002                           RTI
4294
4295                                    ;;*****************************************************************
4296                                    ;HANDLER TO READ CHECK ON DX1.
4297                                    ;WILL GO TO NEXT SECTOR/TRK IF HARD ERROR.
4298                                    ;WILL GO TO EMPTY BUFFER IF NO ERROR OR SOFT ERROR.
4299                                    ;WILL GO TO READ SECTOR (SAME) ON ERROR.
4300                                    ;A RESTORE IS PERFORMED AFTER ANY SEEK ERROR (RNF).
4301                                    ;;*****************************************************************
4302
4303  017524  005737  177170           D1RCHK: TST     RXCS            ;ERROR?
4304  017530  100065                           BPL     1$              ;BR IF NO
4305  017532  023727  003034  000012           CMP     D1ERR,#10.      ;10 ERRORS?
4306  017540  001417                           BEQ     3$              ;BR IF YES
4307  017542  005237  003034                   INC     D1ERR           ;ELSE TRY AGAIN
4308  017546  032737  000020  015240           BIT     #RNF,SRXES      ;SEEK ERROR?
4309  017554  001745                           BEQ     D1RSEC          ;BR IF NO
4310  017556  004737  020612                   JSR     R7,SEKTST       ;SEE IF ON TRACK
4311  017562  012737  017470  015234           MOV     #D1RSEC,D1DISP  ;ELSE DO RESTORE & RET TO RD SEC
4312  017570  012737  000137  177170           MOV     #<RESTOR!IE!DX1>,RXCS
4313  017576  000002                           RTI
4314
4315  017600  004737  014062           3$:     JSR     PC,LD1BAD       ;LOAD BAD TRK/SEC TABLE
4316  017604  032737  000020  015240           BIT     #RNF,SRXES      ;SEEK ERROR?
4317  017612  001417                           BEQ     7$              ;BR IF NO
4318  017614  004737  020612                   JSR     R7,SEKTST       ;SEE IF REALLY ON RIGHT TRACK
4319  017620  005737  003516                   TST     SEKFLG
4320  017624  001422                           BEQ     10$             ;BR IF YES
4321  017626  005037  003516                   CLR     SEKFLG          ;ELSE CLR FLG
4322  017632  104072                           ERROR   72              ;SEEK ERROR READ SECTOR
4323
4324  017634  012737  020520  015234           MOV     #D1NEXT,D1DISP  ;DO RESTORE
4325  017642  012737  000137  177170           MOV     #<RESTOR!IE!DX1>,RXCS
4326  017650  000002                           RTI
4327
4328  017652  032737  000010  015240   7$:     BIT     #CRC,SRXES      ;CRC ERROR?
4329  017660  001404                           BEQ     10$             ;BR IF NO
4330  017662  005237  003052                   INC     D1CRC           ;ELSE SET FLAG
4331  017666  104111                           ERROR   111             ;HARD CRC ERROR
4332  017670  000424                           BR      2$              ;& GO TO EMP BUFF TO CHK DATA
4333
4334  017672  005037  003052           10$:    CLR     D1CRC
4335  017676  104026                           ERROR   26              ;HARD READ ERROR
4336  017700  000137  020520                   JMP     D1NEXT
```

```
4337
4338  017704  005737  003034          1$:     TST     D1ERR           ;ANY PREV ERRORS?
4339  017710  001414                          BEQ     2$              ;BR IF NO
4340  017712  005237  031434                  INC     TSERR           ;TOTAL SOFT ERR CT
4341  017716  005237  031440                  INC     PSERR           ;PASS  SOFT ERR CT
4342  017722  005737  003516                  TST     SEKFLG          ;SEEK ERROR?
4343  017726  001404                          BEQ     9$              ;BR IF NO
4344  017730  005037  003516                  CLR     SEKFLG
4345  017734  104073                          ERROR   73              ;SEEK ERROR READ SECTOR
4346  017736  000401                          BR      2$
4347  017740  104027                  9$:     ERROR   27              ;SOFT ERROR  READ SECTOR
4348
4349  017742  005037  003034          2$:     CLR     D1ERR
4350  017746  000137  017752                  JMP     D1EBUF          ;GO TO EMPTY BUFFER
4351
4352
4353
4354
4355
4356
4357                                  ;;*****************************************************************
4358                                  ; HANDLER TO EMPTY THE DRIVE BUFFER FOR DX1
4359                                  ;& POINT TO THE CHECK DATA HANDLER.
4360                                  ;;*****************************************************************
4361
4362  017752  012703  000100          D1EBUF: MOV     #100,R3                 ;WORD CT
4363  017756  012701  003054                  MOV     #D1BUF,R1               ;BUFFER ADDRESS
4364  017762  012737  020010  015234          MOV     #D1CDAT,D1DISP          ;POINT TO CHK DATA AFT INTER.
4365  017770  012737  000123  177170          MOV     #<EBUF!IE!DX1>,RXCS     ;ISSUE EMPTY BUFFER CMD
4366  017776  013721  177172          1$:     MOV     RXDB,(R1)+              ;GET WORD & STORE IT
4367  020002  005303                          DEC     R3
4368  020004  001374                          BNE     1$
4369  020006  000002                          RTI
4370
```

```
                        ;;*****************************************************************
                        ; 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 ROUTINE IS ALSO ENTERED FROM A CRC ERROR FROM A READ COMMAND.
                        ;;*****************************************************************

020010  005037  003042  D1CDAT: CLR     D1CMER
020014  012703  000100          MOV     #100,R3         ;WORD CTR
020020  012704  003276          MOV     #WCP-2,R4       ;GET TABLE ADDR
020024  013701  003036          MOV     D1TRK,R1
020030  000301                  SWAB    R1
020032  053701  003040          BIS     D1SEC,R1

020036  005737  003510          TST     COMP1           ;DONT SET BIT15 IN COMPATABI' ITY TESTS
020042  001005                  BNE     6$
020044  005737  003512          TST     COMP2
020050  001002                  BNE     6$
020052  052701  100000          BIS     #BIT15,R1       ;TO DISTINGUISH DX1 DATA FROM DX0 DATA

020056  012702  003054  6$:     MOV     #D1BUF,R2       ;GET BUFFER ADDR
020062  020112                  CMP     R1,(R2)         ;1'ST WORD COMPARE OK?
020064  001414                  BEQ     2$              ;BR IF YES
020066  010137  003046          MOV     R1,D1EXP        ;ELSE SAVE
020072  011237  003050  8$:     MOV     (R2),D1REC
020076  005237  003042          INC     D1CMER
020102  000414                  BR      3$

020104  021412          1$:     CMP     (R4),(R2)       ;COMPARE ALL REST
020106  001403                  BEQ     2$              ;BR IF OK
020110  011437  003046          MOV     (R4),D1EXP      ;ELSE SAVE
020114  000766                  BR      8$

020116  005303          2$:     DEC     R3
020120  001405                  BEQ     3$
020122  062702  000002          ADD     #2,R2
020126  062704  000002          ADD     #2,R4
020132  000764                  BR      1$

020134  005737  003042  3$:     TST     D1CMER          ;ANY COMP ERR?
020140  001413                  BEQ     4$              ;BR IF NO
020142  005737  003052          TST     D1CRC           ;HERE FROM CRC ERROR?
020146  001402                  BEQ     14$             ;BR IF NO
020150  104112                  ERROR   112             ;DATA CRC ERROR
020152  000423                  BR      5$
020154  023727  003044  000012  14$:    CMP     D1CERR,#10.     ;ELSE, 10 ERRS YET?
020162  001025                  BNE     7$              ;BR IF NO
020164  104030                  ERROR   30              ;HARD ERROR  DATA COMP
020166  000415                  BR      5$
```

```
4420
4421  020170  005737  003052        4$:     TST     D1CRC           ;HERE FROM CRC ERROR?
4422  020174  001402                        BEQ     15$             ;BR IF NO
4423  020176  104113                        ERROR   113             ;CRC ERR WITH DATA OK
4424  020200  000410                        BR      5$
4425  020202  005737  003044        15$:    TST     D1CERR          ;ANY ERR?
4426  020206  001405                        BEQ     5$
4427  020210  005237  031434                INC     TSERR           ;TOTAL SOFT ERR CT
4428  020214  005237  031440                INC     PSERR           ;PASS  SOFT ERR CT
4429  020220  104031                        ERROR   31              ;SOFT ERROR  DATA COMP
4430
4431  020222  005037  003044        5$:     CLR     D1CERR
4432  020226  005037  003052                CLR     D1CRC
4433  020232  000137  020520                JMP     D1NEXT
4434
4435  020236  005237  003044        7$:     INC     D1CERR
4436  020242  000137  017470                JMP     D1RSEC          ;READ SECTOR OVER AGAIN
4437
4438
4439                                ;;*****************************************************************
4440                                ;HANDLER TO SETUP NEXT SECTOR/TRACK FOR DX1.
4441                                ;WILL GO TO READ SECTOR (NEXT) IF ALL SECTORS ON TRACK NOT DONE.
4442                                ;WILL GO TO FILL BUFFER (NEXT TRACK) IF ALL SECTORS ON TRACK DONE.
4443                                ;WILL SHUT OFF DRV INTERRUPTS IF ALL TRACKS DONE, WITH A MESSAGE.
4444                                ;;*****************************************************************
4445
4446  020246  005037  003034        D1NXT1: CLR     D1ERR
4447  020252  005037  003042                CLR     D1CMER
4448  020256  004537  020532                JSR     R5,NXTSEC       ;CALC NEXT SECTOR
4449  020262  003040                          D1SEC
4450  020264  000137  017470        2$:     JMP     D1RSEC          ;GO READ SECTOR
4451
4452  020270  004737  012624        3$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
4453  020274  000415                            BR  6$              ;RET HERE IF NOT
4454  020276  032777  010000  160634        BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
4455  020304  001411                        BEQ     6$
4456  020306  104401  021577                TYPE    ,DRV1
4457  020312  104401  022504                TYPE    ,TRK
4458  020316  013746  003036                MOV     D1TRK,-(SP)     ;;SAVE D1TRK FOR TYPEOUT
4459  020322  104405                        TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
4460  020324  104401  022574                TYPE    ,DUN
4461  020330  005237  003500        6$:     INC     FIN             ;SETUP TO GO TO DX0
4462  020334  023737  003036  003502        CMP     D1TRK,MAXTRK    ;LAST TRACK?
4463  020342  001406                        BEQ     4$              ;BR IF YES
4464  020344  005237  003036                INC     D1TRK           ;BUMP TRACK
4465  020350  012737  000001  003040        MOV     #1,D1SEC        ;INIT SECTOR
4466  020356  000417                        BR      5$
4467
4468  020360  004737  012624        4$:     JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
4469  020364  000412                            BR  7$              ;RET HERE IF NOT
4470  020366  032777  010000  160544        BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
4471  020374  001406                        BEQ     7$
4472  020376  104401  021577                TYPE    ,DRV1
4473  020402  104401  022511                TYPE    ,PATT
4474  020406  104401  022574                TYPE    ,DUN
4475  020412  005237  003030        7$:     INC     D1PAT           ;PATT DONE FLAG
```

```
 4476  020416  042737  000120  177170  5$:     BIC     #<IE!DX1>,RXCS  ;DISABLE DX1 INTERRUPTS
 4477  020424  000002                          RTI
 4478
 4479
 4480                                  ;;*****************************************************************
 4481                                  ;HANDLER TO SETUP TO GO TO NEXT RANDOM TRACK/SECTOR FOR DX1.
 4482                                  ;;*****************************************************************
 4483
 4484  020426  005037  003034          D1NXT2: CLR     D1ERR
 4485  020432  005037  003042                  CLR     D1CMER
 4486  020436  005237  003500                  INC     FIN             ;SETUP TO GO TO DX0
 4487  020442  005237  003032                  INC     D1CNT
 4488  020446  023737  003032  003504          CMP     D1CNT,MAXSK     ;DID ALL SEEKS?
 4489  020454  001015                          BNE     1$              ;BR IF NO
 4490  020456  004737  012624                  JSR     PC,CHKCON       ;CHECK IF CONSOLE PRESENT
 4491  020462  000412                              BR  1$              ;RET HERE IF NOT
 4492  020464  032777  010000  160446          BIT     #BIT12,@SWR     ;SKIP TYPEOUT IF NOT SET
 4493  020472  001406                          BEQ     1$
 4494  020474  104401  021577                  TYPE    ,DRV1
 4495  020500  104401  022524                  TYPE    ,RANDOM
 4496  020504  104401  022574                  TYPE    ,DUN
 4497  020510  042737  000120  177170  1$:     BIC     #<IE!DX1>,RXCS  ;DISABLE DX1 INTERRUPTS
 4498  020516  000002                          RTI
 4499
 4500
 4501  020520  023727  003476  000001  D1NEXT: CMP     DXTST,#1
 4502  020526  001647                          BEQ     D1NXT1
 4503  020530  000736                          BR      D1NXT2
 4504
 4505
 4506
 4507
 4508                                  ;;*****************************************************************
 4509                                  ;ROUTINE TO DETERMINE NEXT SECTOR FOR INTERLEAVING
 4510                                  ;;*****************************************************************
 4511
 4512  020532  027527  000000  000031  NXTSEC: CMP     @0(R5),#31              ;ARG IS DOSEC/D1SEC
 4513  020540  001003                          BNE     1$
 4514  020542  012735  000002                  MOV     #2,@(R5)+               ;BUMP RET ADDR
 4515  020546  000420                          BR      4$
 4516
 4517  020550  027527  000000  000032  1$:     CMP     @0(R5),#32
 4518  020556  001003                          BNE     2$
 4519  020560  012735  000003                  MOV     #3,@(R5)+
 4520  020564  000411                          BR      4$
 4521
 4522  020566  027527  000000  000030  2$:     CMP     @0(R5),#30
 4523  020574  001403                          BEQ     3$
 4524  020576  062735  000003                  ADD     #3,@(R5)+
 4525  020602  000402                          BR      4$
 4526
 4527  020604  062705  000006          3$:     ADD     #6,R5                   ;SKIP ARG, JMP DOFBUF/D1FBUF
 4528  020610  000205                  4$:     RTS     R5
```

```
                                ;;*****************************************************************
                                ;ROUTINE TO READ MANY SECTORS ON A GIVEN TRACK.
                                ;IF GET RNF ON ALL READS, IT'S ASSUMED THE HEAD IS ON THE WRONG TRACK & SEKFLG SETS.
                                ;IF ANY SECTOR READS CORRECTLY, IT'S ASSUMED THE INITIAL ERROR WAS
                                ;DUE TO READ HEADER ERROR & SEKFLG CLEARS.
                                ;THIS ROUTINE PERFORMS IN FLAG MODE.
                                ;;*****************************************************************

020612  005037  003516          SEKTST: CLR     SEKFLG
020616  042737  000100  177170          BIC     #IE,RXCS        ;TURN OFF INTR ENABLE
020624  105037  177174                  CLRB    RXSA            ;SET UP CURRENT TRK WITH SECTOR 1
020630  005237  177174                  INC     RXSA
020634  052737  000007  177170  4$:     BIS     #RSEC,RXCS      ;ISSUE READ SECTOR
020642  004537  013632                  JSR     R5,TIMER2       ;WAIT FOR DISK DONE
020646  177170                             RXCS
020650  000200                             DONE
020652  104131                          ERROR   131             ;NO DONE
020654  000240                          NOP

020656  032737  000020  177172          BIT     #RNF,RXES       ;RNF ERROR?
020664  001412                          BEQ     2$              ;BR IF NO
020666  122737  000031  177174          CMPB    #31,RXSA        ;LAST SECTOR?
020674  003404                          BLE     3$              ;BR IF YES
020676  062737  000004  177174          ADD     #4,RXSA         ;ELSE TRY NEXT SECTOR
020704  000753                          BR      4$

020706  005237  003516          3$:     INC     SEKFLG          ;SET SEEK ERROR FLAG
020712  000207                  2$:     RTS     R7
```

```
                        ;;*****************************************************************
                        ;THE FOLLOWING CODE, UP TO THE PROGRAM MESSAGES, IS A UTILITY TO COPY
                        ;THE DISKETTE CONTAINING ONLY THE BOOT & THE EXERCISER FROM DX0 TO DX1.
                        ;THE OPERATOR CAN TYPE 'P' TO DO ANOTHER COPY OR
                        ;TYPE '240G' TO ENTER NORMAL TESTING.
                        ;ENTIRE CODE, INCLUDING TEXT & ERROR TABLE, IS 400(10) WORDS.
                        ;;*****************************************************************

020714  005037  003506          COPY:   CLR     COPFLG                  ;DONT NEED FLAG ANY MORE
020720  104401  023023                  TYPE    ,COPMSG                 ;INSERT SCRATCH IN DX1 & TYPE P
020724  000000                          HALT                            ;WAIT FOR PROCEED

020726  004737  021234                  JSR     PC,GETSEC               ;CALCULATE MAX SECTORS TO BE COPIED

020732  005037  021226                  CLR     ERFLG                   ;INITIALIZE
020736  012737  000001  021230          MOV     #1,TRACK
020744  012737  000001  021232          MOV     #1,SECT
020752  105737  177170                  TSTB    RXCS                    ;DONE?  (CTL RDY)
020756  100375                          BPL     .-4                     ;BR IF NO & TRY AGAIN

020760  004537  021304          1$:     JSR     R5,RDWR         ;READ DX0
020764  000007                             RSEC

020766  005737  177170                  TST     RXCS                    ;ERROR?
020772  100011                          BPL     2$                      ;BR IF NO
020774  005237  021226                  INC     ERFLG
021000  023727  021226  000012          CMP     ERFLG,#10.              ;10 ERRORS?
021006  001364                          BNE     1$                      ;BR IF NO & TRY AGAIN
021010  104100                          ERROR   100                     ;HARD ERR READ SECTOR DX0
021012  000000                          HALT
021014  000776                          BR      .-2                     ;FORCE A RESTART FROM 250

021016  005037  021226          2$:     CLR     ERFLG
021022  004537  021344                  JSR     R5,EBUFF        ;EMPTY BUFFER INTO DOBUF
021026  002604                             DOBUF

021030  004537  021304          3$:     JSR     R5,RDWR         ;WRITE DX1
021034  000025                             <WSEC!DX1>

021036  005737  177170                  TST     RXCS                    ;ERROR?
021042  100011                          BPL     4$                      ;BR IF NO
021044  005237  021226                  INC     ERFLG
021050  023727  021226  000012          CMP     ERFLG,#10.              ;10 ERRORS?
021056  001364                          BNE     3$                      ;BR IF NO & TRY AGAIN
021060  104101                          ERROR   101                     ;ERROR WRITE SECTOR
021062  000000                          HALT
021064  000776                          BR      .-2                     ;FORCE RESTART AT 250
```

```
4606
4607  021066  005037  021226          4$:     CLR     ERFLG
4608  021072  004537  021304                  JSR     R5,RDWR         ;READ DX1
4609  021076  000027                             <RSEC!DX1>
4610
4611  021100  005737  177170                  TST     RXCS                    ;ERROR?
4612  021104  100011                          BPL     5$                      ;BR IF NO
4613  021106  005237  021226                  INC     ERFLG
4614  021112  023727  021226  000012          CMP     ERFLG,#10.              ;10 ERRORS?
4615  021120  001362                          BNE     4$                      ;BR IF NO & TRY AGAIN
4616  021122  104103                          ERROR   103                     ;HARD ERR READ SECTOR DX1
4617  021124  000000                          HALT
4618  021126  000776                          BR      .-2                     ;FORCE 250 RESTART
4619
4620  021130  004537  021344          5$:     JSR     R5,EBUFF        ;EMPTY BUFFER INTO D1BUF
4621  021134  003054                             D1BUF
4622
4623  021136  004737  021406                  JSR     PC,DATCHK       ;COMPARE DOBUF WITH D1BUF
4624
4625  021142  005237  021224                  INC     SECCNT                  ;TOTAL DONE
4626
4627  021146  023737  021224  021222          CMP     SECCNT,MAXSEC           ;DID TOTAL # SECTORS?
4628  021154  001415                          BEQ     8$                      ;BR IF YES
4629
4630  021156  023727  021232  000032          CMP     SECT,#32                ;ELSE DID WE DO LAST SECTOR ON TRACK?
4631  021164  001403                          BEQ     6$                      ;BR IF YES
4632  021166  005237  021232                  INC     SECT                    ;ELSE BUMP & DO ANOTHER
4633  021172  000672                          BR      1$
4634
4635  021174  005237  021230          6$:     INC     TRACK                   ;BUMP TRK
4636  021200  012737  000001  021232          MOV     #1,SECT                 ;INIT SECTOR
4637  021206  000664                          BR      1$                      ;& DO ANOTHER
4638
4639  021210  104401  023124          8$:     TYPE    ,COPDUN                 ;DONE MSG
4640  021214  000000                          HALT
4641  021216  000137  020714                  JMP     COPY                    ;DO AGAIN IF 'P' TYPED
4642
4643  021222  000000                  MAXSEC: 0                               ;TOTAL # SECTORS TO READ FROM DX0
4644  021224  000000                  SECCNT: 0                               ;TOTAL # SECTORS WRITTEN TO DX1
4645  021226  000000                  ERFLG:  0
4646  021230  000000                  TRACK:  0
4647  021232  000000                  SECT:   0
```

```
4648
4649                            ;;*****************************************************************
4650                            ;ROUTINE TO CALCULATE THE MAX # OF SECTORS TO READ FROM DX0
4651                            ;;*****************************************************************
4652
4653  021234  012700  032276   GETSEC: MOV     #LSTAD,R0               ;GET MAX ADDR
4654  021240  006200                   ASR     R0                      ;GET WORD CT
4655  021242  005500                   ADC     R0                      ;DONT LOOSE ANY
4656  021244  062700  000377           ADD     #377,R0
4657  021250  042700  000377           BIC     #377,R0                 ;ROUND OFF TO NEAREST WHOLE WORD
4658
4659  021254  000300                   SWAB    R0
4660  021256  006300                   ASL     R0
4661  021260  006300                   ASL     R0                      ;DIVIDE BY 100(8) FOR SECTOR CT.
4662
4663  021262  062700  000074           ADD     #<15.*4>,R0             ;ADD SECTOR CT FOR BLOCKS 0 - 14 OF DX0
4664  021266  062700  000032           ADD     #26.,R0                 ;FACTOR TO TAKE CARE OF RT-11 SECTOR MAPPING
4665
4666  021272  010037  021222           MOV     R0,MAXSEC               ;SAVE
4667                                                                   ;THIS IS THE MAX SECTORS WE WILL READ OFF DX0
4668                                                                   ;& WRITE TO DX1 STARTING AT TRK 1, SECT 1
4669  021276  005037  021224           CLR     SECCNT                  ;COUNT OF TOTAL SECTORS WRITTEN TO DX1
4670  021302  000207                   RTS     PC
4671
4672
4673                            ;;*****************************************************************
4674                            ;ROUTINE TO READ OR WRITE DX0 OR DX1
4675                            ;R5 POINTS TO THE COMMAND & DISK #
4676                            ;;*****************************************************************
4677
4678  021304  013700  021230   RDWR:   MOV     TRACK,R0
4679  021310  000300                   SWAB    R0
4680  021312  053700  021232           BIS     SECT,R0
4681  021316  010037  177174           MOV     R0,RXSA                 ;LOAD TRK/SEC
4682  021322  012537  177170           MOV     (R5)+,RXCS              ;ISSUE COMMAND
4683  021326  004537  013632           JSR     R5,TIMER2               ;WAIT FOR DONE
4684  021332  177170                       RXCS
4685  021334  000200                       DONE
4686  021336  104124                   ERROR   124                     ;NO DONE
4687  021340  000000                   HALT
4688  021342  000205                   RTS     R5
4689
```

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

021344  012700  000100          EBUFF:  MOV     #100,R0                 ;WD CT
021350  012501                          MOV     (R5)+,R1                ;BUFFER ADDR
021352  012737  000003  177170          MOV     #EBUF,RXCS              ;ISSUE EMPTY BUFFER COMMAND
021360  013721  177172          1$:     MOV     RXDB,(R1)+              ;PUT IT IN TABLE
021364  005300                          DEC     R0
021366  001374                          BNE     1$
021370  004537  013632                  JSR     R5,TIMER2               ;WAIT FOR DONE
021374  177170                             RXCS
021376  000200                             DONE
021400  104125                          ERROR   125                     ;NO DONE
021402  000000                          HALT
021404  000205                          RTS     R5


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

021406  012700  000100          DATCHK: MOV     #100,R0                 ;WD CT
021412  012701  002604                  MOV     #D0BUF,R1
021416  012702  003054                  MOV     #D1BUF,R2
021422  022122                  1$:     CMP     (R1)+,(R2)+             ;COMPARE?
021424  001403                          BEQ     2$                      ;BR IF YES
021426  104102                          ERROR   102                     ;MISCOMPARE
021430  000000                          HALT
021432  000776                          BR      .-2

021434  005300                  2$:     DEC     R0
021436  001371                          BNE     1$
021440  000207                          RTS     PC
```

```
  4727
  4728                                  .SBTTL  PROGRAM MESSAGES
  4729
  4730                                  ;;*****************************************************************
  4731                                  ; MESSAGES & ERROR FORMATS
  4732                                  ;;*****************************************************************
  4733
  4734                                  .NLIST  BEX

        021442  041600  052514  052123  CLOPT:  .ASCIZ  <CRLF>/CLUSTER OPTION/
        021462  052200  051105  020115  CLT1:   .ASCIZ  <CRLF>/TERM #1/
        021473     200  042524  046522  CLT2:   .ASCIZ  <CRLF>/TERM #2/
        021504  052200  051105  020115  CLT3:   .ASCIZ  <CRLF>/TERM #3/
        021515     200  054523  041516  SCOM:   .ASCIZ  <CRLF>/SYNC COMM/
        021530  040600  054523  041516  ACOM:   .ASCIZ  <CRLF>/ASYNC COMM/
        021544  050200  044522  052116  PRINT:  .ASCIZ  <CRLF>/PRINTER/
        021555     040  047520  052122  PORT:   .ASCIZ  / PORT TESTED/
        021572  042200  030130     000  DRV0:   .ASCIZ  <CRLF>/DX0/
        021577     200  054104  000061  DRV1:   .ASCIZ  <CRLF>/DX1/
        021604  043200  046111  020114  BUFTST: .ASCIZ  <CRLF>/FILL & EMPTY BUFFER/
        021631     200  052522  047116  NOCON:  .ASCIZ  <CRLF>/RUNNING WITH NO CONSOLE IS SELECTED/
        021676  041600  047514  045503  CLOCK:  .ASCIZ  <CRLF>/CLOCK/
        021705     200  044505  026523  EISTXT: .ASCIZ  <CRLF>/EIS-FIS OPTION/
        021725     113  046440  046505  MEM:    .ASCIZ  /K MEMORY PRESENT/
        021746  005200  051120  043517  DEFBD:  .ASCII  <CRLF><LF>/PROGRAM DEFAULTS (UNLESS MODIFIED):/
        022013     200  041411  052514          .ASCII  <CRLF>/ CLUSTER @ 2400 BAUD/
        022040  004600  051120  047111          .ASCII  <CRLF>/ PRINTER @ 9600 BAUD/
        022065     200  051411  047131          .ASCIZ  <CRLF>/ SYNC & ASYNC COMM PORTS @ 9600 BAUD/

        022133     040  047516  020124  NOTPRES:        .ASCIZ  / NOT PRESENT/
        022150  005200  047111  042523  WARNING:        .ASCII  <CRLF><LF>/INSERT SCRATCH DISKS, TYPE 'P' FOR NORMAL TESTING/
        022233     200  031047  030064          .ASCII  <CRLF>/'240G' FOR NORMAL RESTARTS/
        022266  023600  032462  043460          .ASCII  <CRLF>/'250G' TO COPY SYS EXERCISER DISK/
        022330  023600  033062  043460          .ASCII  <CRLF>/'260G' FOR COMPATABILITY PASS 1: WRITE/
        022377     200  031047  030067          .ASCIZ  <CRLF>/'270G' FOR COMPATABILITY PASS 2: READ/
        022446  052040  051505  044524  DROP:   .ASCIZ  / TESTING DROPPED/
        022467     040  052522  047116  RUN:    .ASCIZ  / RUNNING/
        022500  047440  000113          OK:     .ASCIZ  / OK/
        022504  052040  045522     000  TRK:    .ASCIZ  / TRK/
        022511     040  040504  040524  PATT:   .ASCIZ  / DATA PATT/
        022524  051040  047101  047504  RANDOM: .ASCIZ  / RANDOM SEEKS/
        022542  042600  051511  043055  FISEIS: .ASCIZ  <CRLF>/EIS-FIS TESTS/
        022561     200  042515  020115  MEMORY: .ASCIZ  <CRLF>/MEM TESTS/
        022574  042040  047117  000105  DUN:    .ASCIZ  / DONE/
        022602  005200  040520  051523  PASS:   .ASCIZ  <CRLF><LF>/PASS # /
        022614  051600  051105  040511  SERIAL: .ASCIZ  <CRLF>/SERIAL NO. = /
        022633     200  042412  042116  ENDPAS: .ASCIZ  <CRLF><LF>/END PASS #/
        022650  052011  052117  046101  MSG1:   .ASCIZ  /       TOTAL ERRORS:         /
        022677     200  004411  052011  MSG2:   .ASCIZ  <CRLF>/                    TOTAL SOFT ERRORS:      /
        022733     200  004411  052011  MSG3:   .ASCIZ  <CRLF>/                    TOTAL ERRORS THIS PASS:/
        022767     200  004411  051411  MSG4:   .ASCIZ  <CRLF>/                    SOFT  ERRORS THIS PASS:/
        023023     200  047503  054520  COPMSG: .ASCII  <CRLF>/COPY DX0 TO DX1/
        023043     200  047111  042523          .ASCIZ  <CRLF>/INSERT SCRATCH DISK IN DXi, TYPE 'P' TO PROCEED/
        023124  042200  047117  027105  COPDUN: .ASCIZ  <CRLF>/DONE...TYPE 'P' TO DO AGAIN OR '240G' FOR NORMAL TESTING/
        023216  005200  047503  050115  CC 'AT: .ASCII  <CRLF><LF>/COMPATABILITY PASS 1 (WRITE) DONE/
        023261     200  047504  023440          .ASCIZ  <CRLF>/DO 'P' FOR PASS 2 (READ)/
```

```
023313     200  042504  046526  DEVM:   .ASCIZ  <CRLF>/DEVM = /
023324  042515  047515  054522  EM1:    .ASCIZ  /MEMORY TIMEOUT/
023343     115  046505  042040  EM2:    .ASCIZ  /MEM DATA COMP ERR/
023365     123  047131  020103  EM3:    .ASCIZ  /SYNC COMM DID NOT RECEIVE SYNC CHAR/
023431     120  044522  052116  EM4:    .ASCIZ  /PRINTER STATUS ERR/
023454  042524  046522  021440  EM5:    .ASCIZ  /TERM #1 DATA COMP ERR/
023502  042524  046522  021440  EM6:    .ASCIZ  /TERM #2 DATA COMP ERR/
023530  042524  046522  021440  EM7:    .ASCIZ  /TERM #3 DATA COMP ERR/
023556  051501  047131  020103  EM8:    .ASCIZ  /ASYNC COMM DATA COMP ERR/
023607     123  047131  020103  EM9:    .ASCIZ  /SYNC COMM DATA COMP ERR/
023637     125  042516  050130  EM10:   .ASCIZ  /UNEXP DISK INTERRUPT/
023664  054104  020060  040510  EM11:   .ASCIZ  /DX0 HARD ERR - WRITE SECT/
023716  054104  020060  047523  EM12:   .ASCIZ  /DX0 SOFT ERR - WRITE SECT/
023750  054104  020060  040510  EM13:   .ASCIZ  /DX0 HARD ERR - READ SECT/
024001     104  030130  051440  EM14:   .ASCIZ  /DX0 SOFT ERR - READ SECT/
024032  054104  020060  047125  EM15:   .ASCIZ  /DX0 UNRECOV ERR - DATA COMP/
024066  054104  020060  051105  EM16:   .ASCIZ  /DX0 ERR - DATA COMP/
024112  054104  020061  040510  EM17:   .ASCIZ  /DX1 HARD ERR - WRITE SECT/
024144  054104  020061  047523  EM18:   .ASCIZ  /DX1 SOFT ERR - WRITE SECT/
024176  054104  020061  040510  EM19:   .ASCIZ  /DX1 HARD ERR - READ SECT/
024227     104  030530  051440  EM20:   .ASCIZ  /DX1 SOFT ERR - READ SECT/
024260  054104  020061  047125  EM21:   .ASCIZ  /DX1 UNRECOV ERR - DATA COMP/
024314  054104  020061  051105  EM22:   .ASCIZ  /DX1 ERR - DATA COMP/
024340  046103  020113  052510  EM23:   .ASCIZ  /CLK HUNG/
024351     120  044522  052116  EM24:   .ASCIZ  /PRINTER HUNG/
024366  042524  046522  021440  EM25:   .ASCIZ  /TERM #1 HUNG/
024403     124  051105  020115  EM26:   .ASCIZ  /TERM #2 HUNG/
024420  042524  046522  021440  EM27:   .ASCIZ  /TERM #3 HUNG/
024435     123  047131  020103  EM28:   .ASCIZ  /SYNC COMM HUNG/
024454  051501  047131  020103  EM29:   .ASCIZ  /ASYNC COMM HUNG/
024474  054104  020060  052510  EM30:   .ASCIZ  /DX0 HUNG/
024505     104  030530  044040  EM31:   .ASCIZ  /DX1 HUNG/
024516  054523  041516  041440  EM32:   .ASCIZ  /SYNC COMM - DATA NOT AVAIL NOT SET/
024561     123  047131  020103  EM33:   .ASCIZ  /SYNC COMM - RECVR ACTIVE NOT SET/
024622  054523  041516  041440  EM34:   .ASCIZ  /SYNC COMM - RECVR DONE NOT SET/
024661     104  030130  023440  EM35:   .ASCIZ  /DX0 'DEL DATA' BIT 5 NOT SET IN RXES/
024726  054104  020060  047111  EM36:   .ASCIZ  /DX0 INIT CMD DID NOT READ TRK 1, SEC 1/
024775     104  030130  051040  EM37:   .ASCIZ  /DX0 RESTORE CMD ERR/
025021     104  030130  023440  EM38:   .ASCIZ  /DX0 'INIT DONE' BIT 0 NOT SET IN RXES/
025067     104  030130  051040  EM39:   .ASCIZ  /DX0 READ STATUS CMD ERR/
025117     104  030130  044440  EM40:   .ASCIZ  /DX0 INV ADDR BIT 1 NOT SET IN RXES/
025162  054104  020060  051105  EM41:   .ASCIZ  /DX0 ERR BIT NOT SET IN RXCS/
025216  054104  020060  047111  EM42:   .ASCIZ  /DX0 INIT CMD ERR/
025237     104  030130  052440  EM43:   .ASCIZ  /DX0 UNRECOV SEEK ERR - WRITE SECT/
025301     104  030130  051040  EM44:   .ASCIZ  /DX0 RECOV SEEK ERR - WRITE SECT/
025341     104  030130  052440  EM45:   .ASCIZ  /DX0 UNRECOV SEEK ERR - READ SECT/
025402  054104  020060  042522  EM46:   .ASCIZ  /DX0 RECOV SEEK ERR - READ SECT/
025441     104  030530  052440  EM47:   .ASCIZ  /DX1 UNRECOV SEEK ERR - WRITE SECT/
025503     104  030530  051040  EM48:   .ASCIZ  /DX1 RECOV SEEK ERR - WRITE SECT/
025543     104  030530  052440  EM49:   .ASCIZ  /DX1 UNRECOV SEEK ERR - READ SECT/
025604  054104  020061  042522  EM50:   .ASCIZ  /DX1 RECOV SEEK ERR - READ SECT/
025643     104  030530  051040  EM51:   .ASCIZ  /DX1 RESTORE CMD ERR/
025667     104  051511  020113  EM52:   .ASCIZ  /DISK 'DONE' NOT SET/
025713     106  046111  020114  EM53:   .ASCIZ  /FILL & EMPTY BUFF TEST... DATA MISCOMP/
025762  040504  040524  046440  EM54:   .ASCIZ  /DATA MISCOMP/
025777     122  050105  040514  EM55:   .ASCIZ  /REPLACE DX0...10 BAD SECTORS/
```

```
026034  042522  046120  041501  EM56:   .ASCIZ  /REPLACE DX1...10 BAD SECTORS/
026071     104  030130  044040  EM57:   .ASCIZ  /DX0 HARD CRC ERR - READ SECT/
026126  054104  020060  051120  EM58:   .ASCIZ  /DX0 PREV CRC ERR WITH DATA WRONG/
026167     104  030130  050040  EM59:   .ASCIZ  /DX0 PREV CRC ERR WITH DATA OK/
026225     104  030530  044040  EM60:   .ASCIZ  /DX1 HARD CRC ERR - READ SECT/
026262  054104  020061  051120  EM61:   .ASCIZ  /DX1 PREV CRC ERR WITH DATA WRONG/
026323     104  030530  050040  EM62:   .ASCIZ  /DX1 PREV CRC ERR WITH DATA OK/
026361     105  051511  043055  EM63:   .ASCIZ  /EIS-FIS CONDITION CODE ERROR/
026416  044505  026523  044506  EM64:   .ASCIZ  /EIS-FIS DATA ERROR/

026441     105  051122  051117  DH1:    .ASCIZ  /ERROR #        ERR PC/
026460  051105  047522  020122  DH2:    .ASCIZ  /ERROR #        ERR PC  ADDR/
026504  051105  047522  020122  DH3:    .ASCIZ  /ERROR #        ERR PC  ADDR    EXPECT  RECVD/
026546  051105  047522  020122  DH4:    .ASCIZ  /ERROR #        ERR PC  PR STATUS/
026577     105  051122  051117  DH5:    .ASCIZ  /ERROR #        ERR PC  EXPECT  RECVD/
026634  051105  047522  020122  DH6:    .ASCIZ  /ERROR #        DXTST#  ERR PC  RXCS    RXES    RXSA/
026701     105  051122  051117  DH7:    .ASCIZ  /ERROR #        DXTST#  ERR PC  RXCS    RXES    TRACK   SECTOR  # RETRIE
026770  051105  047522  020122  DH8:    .ASCIZ  /ERROR #        DXTST#  ERR PC  RXCS    RXSA    EXPECT  RECVD  # RETRIES

        027062                  .EVEN

027062  031430  001116  000000  DT1:    .WORD   ERRNUM,$ERRPC,0
027070  031430  001116  005052  DT2:    .WORD   ERRNUM,$ERRPC,ADDR,0
027100  031430  001116  005052  DT3:    .WORD   ERRNUM,$ERRPC,ADDR,PAT,MEMHLD,0
027114  031430  001116  014334  DT4:    .WORD   ERRNUM,$ERRPC,SPRS,0
027124  031430  001116  014432  DT5:    .WORD   ERRNUM,$ERRPC,T1CHR,T1HLD,0
027136  031430  001116  014532  DT6:    .WORD   ERRNUM,$ERRPC,T2CHR,T2HLD,0
027150  031430  001116  014632  DT7:    .WORD   ERRNUM,$ERRPC,T3CHR,T3HLD,0
027162  031430  001116  014732  DT8:    .WORD   ERRNUM,$ERRPC,COMCHR,COMHLD,0
027174  031430  003476  001116  DT9:    .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,SRXSA,0
027212  031430  003476  001116  DT10:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D0TRK,D0SEC,D0ERR,0
027234  031430  003476  001116  DT11:   .WORD   ERRNUM,DXTST,$ERRPC,RXCS,RXSA,D0EXP,D0REC,D0CERR,0
027256  031430  003476  001116  DT12:   .WORD   ERRNUM,DXTST,$ERRPC,SRXCS,SRXES,D1TRK,D1SEC,D1ERR,0
027300  031430  003476  001116  DT13:   .WORD   ERRNUM,DXTST,$ERRPC,RXCS,RXSA,D1EXP,D1REC,D1CERR,0
027322  031430  003476  001116  DT14:   .WORD   ERRNUM,DXTST,$ERRPC,RXCS,RXES,RXSA,0
027340  031430  001116  007744  DT15:   .WORD   ERRNUM,$ERRPC,EXP0,EBHLD,0
027352  031430  001116  007746  DT16:   .WORD   ERRNUM,$ERRPC,EXP1,EBHLD,0

027364  000000                  DF:     .WORD   0               ;OCTAL FORMAT FOR ALL
027366  000000                          .WORD   0
027370  000000                          .WORD   0

027372  000000                  DF1:    .WORD   0               ;OCTAL,OCTAL
027374  000000                          .WORD   0               ;OCTAL,OCTAL
027376  000400                          .WORD   400             ;DECIMAL,OCTAL
027400  000401                          .WORD   401             ;DECIMAL,DECIMAL

027402  000000                  DF2:    .WORD   0               ;OCTAL,OCTAL
027404  000000                          .WORD   0               ;OCTAL,OCTAL
027406  000000                          .WORD   0               ;OCTAL,OCTAL
027410  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
                                        ;*

027412  105737  001157          $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
027416  100002                          BPL     1$              ;;BR IF YES
027420  000000                          HALT                    ;;HALT HERE IF NO TERMINAL
027422  000430                          BR      3$              ;;LEAVE
027424  010046                  1$:     MOV     R0,-(SP)        ;;SAVE R0
027426  017600  000002                  MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
027432  122737  000001  001210          CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
027440  001011                          BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
027442  132737  000100  001211          BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
027450  001405                          BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
027452  010037  027462                  MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
027456  004737  030544                  JSR     PC,$ATY3        ;;SPOOL MESSAGE TO APT
027462  000000                  61$:    .WORD   0               ;;MESSAGE ADDRESS
027464  132737  000040  001211  62$:    BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
027472  001003                          BNE     60$             ;;YES,SKIP TYPE OUT
027474  112046                  2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
027476  001005                          BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
027500  005726                          TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
027502  012600                  60$:    MOV     (SP)+,R0        ;;RESTORE R0
027504  062716  000002          3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
027510  000002                          RTI                     ;;RETURN
027512  122716  000011          4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
027516  001430                          BEQ     8$
027520  122716  000200                  CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
027524  001006                          BNE     5$
027526  005726                          TST     (SP)+           ;;POP  <CR><LF> EQUIV
027530  104401                          TYPE                    ;;TYPE A CR AND LF
027532  001165                          $CRLF
027534  105037  027670                  CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
027540  000755                          BR      2$              ;;GET NEXT CHARACTER
027542  004737  027624          5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
027546  123726  001156          6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
027552  001350                          BNE     2$              ;;IF NO GO GET NEXT CHAR.
027554  013746  001154                  MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                                ;;AND THE NULL CHAR.
027560  105366  000001          7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
027564  002770                          BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
027566  004737  027624                  JSR     PC,$TYPEC       ;;GO TYPE A NULL
027572  105337  027670                  DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
```

```
 4791  027576  000770                    BR      7$              ;;LOOP
 4792
 4793                                    ;HORIZONTAL TAB PROCESSOR
 4794
 4795  027600  112716  000040            8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
 4796  027604  004737  027624            9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
 4797  027610  132737  000007  027670            BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
 4798  027616  001372                            BNE     9$              ;;TAB STOP
 4799  027620  005726                            TST     (SP)+           ;;POP SPACE OFF STACK
 4800  027622  000724                            BR      2$              ;;GET NEXT CHARACTER
 4801  027624  105777  151320            $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
 4802  027630  100375                            BPL     $TYPEC
 4803  027632  116677  000002  151312            MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
 4804  027640  122766  000015  000002            CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
 4805  027646  001003                            BNE     1$              ;;BRANCH IF NO
 4806  027650  105037  027670                    CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
 4807  027654  000406                            BR      $TYPEX          ;;EXIT
 4808  027656  122766  000012  000002  1$:      CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
 4809  027664  001402                            BEQ     $TYPEX          ;;BRANCH IF YES
 4810  027666  105227                            INCB    (PC)+           ;;COUNT THE CHARACTER
 4811  027670  000000            $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
 4812  027672  000207            $TYPEX: RTS     PC
 4813
```

```
                                .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.
027674  022737  000176  001140  $CKSWR: CMP     #SWREG,SWR      ;;IS THE SOFT-SWR SELECTED?
027702  001114                        BNE     15$             ;;BRANCH IF NO
027704  105777  151234                TSTB    @$TKS           ;;CHAR THERE?
027710  100111                        BPL     15$             ;;IF NO, DON'T WAIT AROUND
027712  117746  151230                MOVB    @$TKB,-(SP)     ;;SAVE THE CHAR
027716  042716  177600                BIC     #^C177,(SP)     ;;STRIP-OFF THE ASCII
027722  022726  000007                CMP     #7,(SP)+        ;;IS IT A CONTROL G?
027726  001102                        BNE     15$             ;;NO, RETURN TO USER
027730  123727  001134  000001        CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
027736  001476                        BEQ     15$             ;;BRANCH IF YES

027740  104401  030506                TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G (^G)
027744  104401  030513        $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
027750  013746  000176                MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
027754  104402                        TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
027756  104401  030524                TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
027762  005046                19$:    CLR     -(SP)           ;;CLEAR COUNTER
027764  005046                        CLR     -(SP)           ;;THE NEW SWR
027766  105777  151152        7$:     TSTB    @$TKS           ;;CHAR THERE?
027772  100375                        BPL     7$              ;;IF NOT TRY AGAIN

027774  117746  151146                MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
030000  042716  177600                BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII

030004  021627  000003                CMP     (SP),#3         ;;IS IT A CONTROL-C?
030010  001015                        BNE     9$              ;;BRANCH IF NOT
030012  104401  030474                TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C (^C)
030016  062706  000006                ADD     #6,SP           ;;CLEAN UP STACK
030022  123727  001135  000001        CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
030030  001003                        BNE     8$              ;;BRANCH IF NO
030032  012777  000100  151104        MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
030040  000137  013126          8$:   JMP     GT$DEV          ;;CONTROL-C RESTART

030044  021627  000025          9$:   CMP     (SP),#25        ;;IS IT A CONTROL-U?
030050  001005                        BNE     10$             ;;BRANCH IF NOT
030052  104401  030501                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U (^U)
030056  062706  000006          20$:  ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
030062  000737                        BR      19$             ;;LET'S TRY IT AGAIN

030064  021627  000015          10$:  CMP     (SP),#15        ;;IS IT A <CR>?
030070  001022                        BNE     16$             ;;BRANCH IF NO
030072  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
030076  001403                        BEQ     11$             ;;BRANCH IF YES
030100  016677  000002  151032        MOV     2(SP),@SWR      ;;SAVE NEW SWR
```

```
 4870  030106  062706  000006          11$:    ADD     #6,SP           ;;CLEAR UP STACK
 4871  030112  104401  001165          14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
 4872  030116  123727  001135  000001          CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
 4873  030124  001003                          BNE     15$             ;;BRANCH IF NOT
 4874  030126  012777  000100  151010          MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
 4875  030134  000002                  15$:    RTI                     ;;RETURN
 4876  030136  004737  027624          16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
 4877  030142  021627  000060                  CMP     (SP),#60        ;;CHAR < 0?
 4878  030146  002420                          BLT     18$             ;;BRANCH IF YES
 4879  030150  021627  000067                  CMP     (SP),#67        ;;CHAR > 7?
 4880  030154  003015                          BGT     18$             ;;BRANCH IF YES
 4881  030156  042726  000060                  BIC     #60,(SP)+       ;;STRIP-OFF ASCII
 4882  030162  005766  000002                  TST     2(SP)           ;;IS THIS THE FIRST CHAR
 4883  030166  001403                          BEQ     17$             ;;BRANCH IF YES
 4884  030170  006316                          ASL     (SP)            ;;NO, SHIFT PRESENT
 4885  030172  006316                          ASL     (SP)            ;;   CHAR OVER TO MAKE
 4886  030174  006316                          ASL     (SP)            ;;   ROOM FOR NEW ONE.
 4887  030176  005266  000002          17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
 4888  030202  056616  177776                  BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
 4889  030206  000667                          BR      7$              ;;GET THE NEXT ONE
 4890  030210  104401  001164          18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
 4891  030214  000720                          BR      20$             ;;SIMULATE CONTROL-U
 4892                                  .DSABL  LSB
 4893
 4894
 4895                                  ;;*****************************************************************
 4896                                  ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
 4897                                  ;*CALL:
 4898                                  ;*      RDCHR                   ;;INPUT A SINGLE CHARACTER FROM THE TTY
 4899                                  ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
 4900                                  ;*                              ;;WITH PARITY BIT STRIPPED OFF
 4901                                  ;
 4902
 4903  030216  011646          $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC
 4904  030220  016666  000004  000002          MOV     4(SP),2(SP)     ;;SAVE THE PS
 4905  030226  105777  150712          1$:     TSTB    @$TKS           ;;WAIT FOR
 4906  030232  100375                          BPL     1$              ;;A CHARACTER
 4907  030234  117766  150706  000004          MOVB    @$TKB,4(SP)     ;;READ THE TTY
 4908  030242  042766  177600  000004          BIC     #^C<177>,4(SP)  ;;GET RID OF JUNK IF ANY
 4909  030250  026627  000004  000023          CMP     4(SP),#23       ;;IS IT A CONTROL-S?
 4910  030256  001013                          BNE     3$              ;;BRANCH IF NO
 4911  030260  105777  150660          2$:     TSTB    @$TKS           ;;WAIT FOR A CHARACTER
 4912  030264  100375                          BPL     2$              ;;LOOP UNTIL ITS THERE
 4913  030266  117746  150654                  MOVB    @$TKB,-(SP)     ;;GET CHARACTER
 4914  030272  042716  177600                  BIC     #^C177,(SP)     ;;MAKE IT 7-BIT ASCII
 4915  030276  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
 4916  030302  001366                          BNE     2$              ;;IF NOT DISCARD IT
 4917  030304  000750                          BR      1$              ;;YES, RESUME
 4918  030306  026627  000004  000140  3$:     CMP     4(SP),#140      ;;IS IT UPPER CASE?
 4919  030314  002407                          BLT     4$              ;;BRANCH IF YES
 4920  030316  026627  000004  000175          CMP     4(SP),#175      ;;IS IT A SPECIAL CHAR?
 4921  030324  003003                          BGT     4$              ;;BRANCH IF YES
 4922  030326  042766  000040  000004          BIC     #40,4(SP)       ;;MAKE IT UPPER CASE
 4923  030334  000002                  4$:     RTI                     ;;GO BACK TO USER
 4924                                  ;;*****************************************************************
 4925                                  ;*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

030336  010346                 $RDLIN: MOV     R3,-(SP)                ;;SAVE R3
030340  012703  030464           1$:    MOV     #$TTYIN,R3              ;;GET ADDRESS
030344  022703  030474           2$:    CMP     #$TTYIN+8.,R3           ;;BUFFER FULL?
030350  101415                         BLOS    4$                      ;;BR IF YES
030352  104410                         RDCHR                           ;;GO READ ONE CHARACTER FROM THE TTY
030354  112613                         MOVB    (SP)+,(R3)              ;;GET CHARACTER
030356  122713  000003                 CMPB    #3,(R3)                 ;;IS IT A CONTROL-C?
030362  001005                         BNE     10$                     ;;BRANCH IF NO
030364  104401  030474                 TYPE    ,$CNTLC                 ;;TYPE A CONTROL-C (^C)
030370  012603                         MOV     (SP)+,R3                ;;RESTORE R3
030372  000137  013126                 JMP     GT$DEV                  ;;GOTO CONTROL-C RESTART
030376  122713  000177           10$:  CMPB    #177,(R3)               ;;IS IT A RUBOUT
030402  001003                         BNE     3$                      ;;SKIP IF NOT
030404  104401  001164           4$:   TYPE    ,$QUES                  ;;TYPE A '?'
030410  000753                         BR      1$                      ;;CLEAR THE BUFFER AND LOOP
030412  111337  030462           3$:   MOVB    (R3),9$                 ;;ECHO THE CHARACTER
030416  104401  030462                 TYPE    ,9$
030422  122723  000015                 CMPB    #15,(R3)+               ;;CHECK FOR RETURN
030426  001346                         BNE     2$                      ;;LOOP IF NOT RETURN
030430  105063  177777                 CLRB    -1(R3)                  ;;CLEAR RETURN (THE 15)
030434  104401  001166                 TYPE    ,$LF                    ;;TYPE A LINE FEED
030440  012603                         MOV     (SP)+,R3                ;;RESTORE R3
030442  011646                         MOV     (SP),-(SP)              ;;ADJUST THE STACK AND PUT ADDRESS OF THE
030444  016666  000004  000002         MOV     4(SP),2(SP)             ;;      FIRST ASCII CHARACTER ON IT
030452  012766  030464  000004         MOV     #$TTYIN,4(SP)
030460  000002                         RTI                             ;;RETURN
030462     000                 9$:     .BYTE   0                       ;;STORAGE FOR ASCII CHAR. TO TYPE
030463     000                         .BYTE   0                       ;;TERMINATOR
030464  000010              $TTYIN:    .BLKB   8.                      ;;RESERVE 8 BYTES FOR TTY INPUT
030474  041536  005015     $CNTLC:    .ASCIZ  /^C/<15><12>            ;;CONTROL "C"
030501     136  006525  000012  $CNTLU:    .ASCIZ  /^U/<15><12>       ;;CONTROL "U"
030506  043536  005015     $CNTLG:    .ASCIZ  /^G/<15><12>            ;;CONTROL "G"
030513     015  051412  051127  $MSWR:     .ASCIZ  <15><12>/SWR = /
030520  036440  000040
030524  020040  042516  020127  $MNEW:     .ASCIZ  /  NEW = /
030532  020075     000
                030536                  .EVEN
```

```
                        .SBTTL  APT COMMUNICATIONS ROUTINE
                        ;;*****************************************************************
030536 112737 000001 031002 $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
030544 112737 000001 031000 $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
030552 000403               BR      $ATYC
030554 112737 000001 031002 $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
030562              $ATYC:
030562 010046               MOV     R0,-(SP)        ;;PUSH R0 ON STACK
030564 010146               MOV     R1,-(SP)        ;;PUSH R1 ON STACK
030566 105737 031000        TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
030572 001450               BEQ     5$              ;;IF NOT:  BR
030574 122737 000001 001210 CMPB    #APTENV,$ENV    ;;OPERATING UNDER APT?
030602 001031               BNE     3$              ;;IF NOT:  BR
030604 132737 000100 001211 BITB    #APTSPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
030612 001425               BEQ     3$              ;;IF NOT:  BR
030614 017600 000004        MOV     @4(SP),R0       ;;GET MESSAGE ADDR.
030620 062766 000002 000004 ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
030626 005737 001170        1$:     TST     $MSGTYPE        ;;SEE IF DONE W/ LAST XMISSION?
030632 001375               BNE     1$              ;;IF NOT:  WAIT
030634 010037 001204        MOV     R0,$MSGAD       ;;PUT ADDR IN MAILBOX
030640 105720               2$:     TSTB    (R0)+           ;;FIND END OF MESSAGE
030642 001376               BNE     2$
030644 163700 001204        SUB     $MSGAD,R0       ;;SUB START OF MESSAGE
030650 006200               ASR     R0              ;;GET MESSAGE LNGTH IN WORDS
030652 010037 001206        MOV     R0,$MSGLGT      ;;PUT LENGTH IN MAILBOX
030656 012737 000004 001170 MOV     #4,$MSGTYPE     ;;TELL APT TO TAKE MSG.
030664 000413               BR      5$
030666 017637 000004 030712 3$:     MOV     @4(SP),4$       ;;PUT MSG ADDR IN JSR LINKAGE
030674 062766 000002 000004 ADD     #2,4(SP)                ;;BUMP RETURN ADDRESS
030702 013746 177776        MOV     177776,-(SP)    ;;PUSH 177776 ON STACK
030706 004737 027412        JSR     PC,$TYPE        ;;CALL TYPE MACRO
030712 000000               4$:     .WORD   0
030714              5$:
030714 105737 031002        10$:    TSTB    $FFLG           ;;SHOULD REPORT FATAL ERROR?
030720 001416               BEQ     12$             ;;IF NOT:  BR
030722 005737 001210        TST     $ENV            ;;RUNNING UNDER APT?
030726 001413               BEQ     12$             ;;IF NOT:  BR
030730 005737 001170        11$:    TST     $MSGTYPE        ;;FINISHED LAST MESSAGE?
030734 001375               BNE     11$             ;;IF NOT:  WAIT
030736 017637 000004 001172 MOV     @4(SP),$FATAL   ;;GET ERROR #
030744 062766 000002 000004 ADD     #2,4(SP)                ;;BUMP RETURN ADDR.
030752 005237 001170        INC     $MSGTYPE        ;;TELL APT TO TAKE ERROR
030756 105037 031002        12$:    CLRB    $FFLG           ;;CLEAR FATAL FLAG
030762 105037 031001        CLRB    $LFLG           ;;CLEAR LOG FLAG
030766 105037 031000        CLRB    $MFLG           ;;CLEAR MESSAGE FLAG
030772 012601               MOV     (SP)+,R1        ;;POP STACK INTO R1
030774 012600               MOV     (SP)+,R0        ;;POP STACK INTO R0
030776 000207               RTS     PC              ;;RETURN
031000    000               $MFLG:  .BYTE   0               ;;MESSG. FLAG
031001    000               $LFLG:  .BYTE   0               ;;LOG FLAG
031002    000               $FFLG:  .BYTE   0               ;;FATAL FLAG
          031004                    .EVEN
          000200            APTSIZE=200
          000001            APTENV=001
          000100            APTSPOOL=100
```

```
                        000040                  APTCSUP=040
                                                .SBTTL  ERROR HANDLER ROUTINE
                                                ;;*****************************************************************
                                                ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
                                                ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
                                                ;*AND GO TO MYTYPE ON ERROR
                                                ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                                ;*SW15=1         HALT ON ERROR
                                                ;*SW13=1         INHIBIT ERROR TYPEOUTS
                                                ;*SW10=1         BELL ON ERROR
                                                ;*CALL
                                                ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

031004                                          $ERROR:
031004  104407                                         CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
031006  105237  001103                         7$:     INCB    $ERFLG                  ;;SET THE ERROR FLAG
031012  001775                                         BEQ     7$                      ;;DON'T LET THE FLAG GO TO ZERO
031014  013777  001102  150120                         MOV     $TSTNM,@DISPLAY         ;;DISPLAY TEST NUMBER AND ERROR FLAG
031022  032777  002000  150110                         BIT     #BIT10,@SWR             ;;BELL ON ERROR?
031030  001402                                         BEQ     1$                      ;;NO - SKIP
031032  104401  001160                                 TYPE    ,$BELL                  ;;RING BELL
031036  005237  001112                         1$:     INC     $ERTTL                  ;;COUNT THE NUMBER OF ERRORS
031042  011637  001116                                 MOV     (SP),$ERRPC             ;;GET ADDRESS OF ERROR INSTRUCTION
031046  162737  000002  001116                         SUB     #2,$ERRPC
031054  117737  150036  001114                         MOVB    @$ERRPC,$ITEMB          ;;STRIP AND SAVE THE ERROR ITEM CODE
031062  032777  020000  150050                         BIT     #BIT13,@SWR             ;;SKIP TYPEOUT IF SET
031070  001004                                         BNE     20$                     ;;SKIP TYPEOUTS
031072  004737  031320                                 JSR     PC,MYTYPE               ;;GO TO USER ERROR ROUTINE
031076  104401  001165                                 TYPE    ,$CRLF
031102                                          20$:
031102  122737  000001  001210                         CMPB    #APTENV,$ENV            ;;RUNNING IN APT MODE
031110  001007                                         BNE     2$                      ;;NO,SKIP APT ERROR REPORT
031112  113737  001114  031124                         MOVB    $ITEMB,21$              ;;SET ITEM NUMBER AS ERROR NUMBER
031120  004737  030554                                 JSR     PC,$ATY4                ;;REPORT FATAL ERROR TO APT
031124     000                                 21$:    .BYTE   0
031125     000                                         .BYTE   0
031126  000777                                 22$:    BR      22$                     ;;APT ERROR LOOP
031130  005777  150004                         2$:     TST     @SWR                    ;;HALT ON ERROR
031134  100002                                         BPL     3$                      ;;SKIP IF CONTINUE
031136  000000                                         HALT                            ;;HALT ON ERROR!
031140  104407                                         CKSWR                           ;;TEST FOR CHANGE IN SOFT-SWR
031142                                          3$:
031142  000002                                         RTI                             ;;RETURN
```

```
                        .SBTTL  ERROR MESSAGE TYPEOUT ROUTINE

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

031144                  $ERRTYP:
031144  104401  001163          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031150  010046                  MOV     R0,-(SP)        ;;SAVE R0
031152  005000                  CLR     R0              ;;PICKUP THE ITEM INDEX
031154  153700  001114          BISB    @#$ITEMB,R0
031160  001004                  BNE     1$              ;;IF ITEM NUMBER IS ZERO, JUST
                                                        ;;TYPE THE PC OF THE ERROR
031162  013746  001116          MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                        ;;ERROR ADDRESS
031166  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
031170  000445                  BR      10$             ;;GET OUT
031172  005300          1$:     DEC     R0              ;;ADJUST THE INDEX SO THAT IT WILL
031174  006300                  ASL     R0              ;;      WORK FOR THE ERROR TABLE
031176  006300                  ASL     R0
031200  006300                  ASL     R0
031202  062700  001250          ADD     #$ERRTB,R0      ;;FORM TABLE POINTER
031206  012037  031216          MOV     (R0)+,2$        ;;PICKUP "ERROR MESSAGE" POINTER
031212  001404                  BEQ     3$              ;;SKIP TYPEOUT IF NO POINTER
031214  104401                  TYPE                    ;;TYPE THE "ERROR MESSAGE"
031216  000000          2$:     .WORD   0               ;;"ERROR MESSAGE" POINTER GOES HERE
031220  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031224  012037  031234  3$:     MOV     (R0)+,4$        ;;PICKUP "DATA HEADER" POINTER
031230  001404                  BEQ     5$              ;;SKIP TYPEOUT IF 0
031232  104401                  TYPE                    ;;TYPE THE "DATA HEADER"
031234  000000          4$:     .WORD   0               ;;"DATA HEADER" POINTER GOES HERE
031236  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031242  010146          5$:     MOV     R1,-(SP)        ;;SAVE R1
031244  012001                  MOV     (R0)+,R1        ;;PICKUP "DATA TABLE" POINTER
031246  001415                  BEQ     9$              ;;BR IF NO DATA TO BE TYPED
031250  012000                  MOV     (R0)+,R0        ;;PICKUP "DATA FORMAT" POINTER
031252  105720          6$:     TSTB    (R0)+           ;;"OCTAL" OR "DECIMAL"
031254  001003                  BNE     7$              ;;BR IF DECIMAL
031256  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
031260  104402                  TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
031262  000402                  BR      8$
031264                  7$:
031264  013146                  MOV     @(R1)+,-(SP)    ;;SAVE @(R1)+ FOR TYPEOUT
031266  104405                  TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
031270  005711          8$:     TST     (R1)            ;;IS THERE ANOTHER NUMBER?
031272  001403                  BEQ     9$              ;;BR IF NO
031274  104401  031314          TYPE    ,11$            ;;TYPE TWO(2) SPACES
031300  000764                  BR      6$              ;;LOOP

031302  012601          9$:     MOV     (SP)+,R1        ;;RESTORE R1
031304  012600          10$:    MOV     (SP)+,R0        ;;RESTORE R0
031306  104401  001165          TYPE    ,$CRLF          ;;"CARRIAGE RETURN" & "LINE FEED"
031312  000207                  RTS     PC              ;;RETURN
031314  020040      000 11$:    .ASCIZ  /  /            ;;TWO(2) SPACES
031320                          .EVEN
```

```
5124
5125
5126
5127  031320  113737  001114  031430  MYTYPE: MOVB    $ITEMB,ERRNUM   ;FOR ERROR TYPEOUT
5128  031326  013737  015242  001174          MOV     SRXSA,$TESTN    ;FOR APT
5129                                                                  ;APT PRINTS $TESTN & $FATAL IN THAT ORDER.
5130                                                                  ;$TESTN WILL BE FLOPPY RXSA
5131                                                                  ;$FATAL IS ERR #
5132  031334  032737  000020  001246          BIT     #BIT4,$DEVM     ;RUNNING WITHOUT CONSOLE?
5133  031342  001407                          BEQ     12$             ;BR IF NO
5134  031344  012737  040300  177420          MOV     #<ULITE!300>,PARAM      ;ELSE SET USER LIGHT 1 & 2
5135  031352  000000                          HALT                            ;AND HALT....WAIT TO PROCEED
5136  031354  012737  040000  17?420          MOV     #<ULITE>,PARAM          ;CLEAR LIGHTS
5137
5138  031362  104401  022602          12$:    TYPE    ,PASS           ;TYPE PASS # BEFORE ERROR
5139  031366  013746  001176          MOV     $PASS,-(SP)     ;LOAD PASS #
5140  031372  005216                  INC     (SP)            ;NO SUCH THING AS PASS 0 FOR OPERATOR
5141  031374  104405                  TYPDS                   ;TYPE IT
5142
5143  031376  104401  022614          TYPE    ,SERIAL         ;TYPE SERIAL # BEFORE ERROR
5144  031402  104401  030464          TYPE    ,$TTYIN
5145  031406  104401  012102          TYPE    ,NULL
5146
5147  031412  004737  031144          JSR     PC,$ERRTYP      ;TYPE ERROR MSG
5148  031416  005237  031432          INC     TERR
5149  031422  005237  031436          INC     PERR
5150  031426  000207                  RTS     PC
5151
5152  031430  000000          ERRNUM: 0                       ;FOR ERR TYPEOUT
5153
5154  031432  000000          TERR:   0                       ;TOTAL ERROR COUNT
5155  031434  000000          TSERR:  0                       ;TOTAL SOFT ERR COUNT
5156  031436  000000          PERR:   0                       ;PASS ERR COUNT
5157  031440  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

031442                          $TYPDS:
031442  010046                         MOV     R0,-(SP)        ;;PUSH R0 ON STACK
031444  010146                         MOV     R1,-(SP)        ;;PUSH R1 ON STACK
031446  010246                         MOV     R2,-(SP)        ;;PUSH R2 ON STACK
031450  010346                         MOV     R3,-(SP)        ;;PUSH R3 ON STACK
031452  010546                         MOV     R5,-(SP)        ;;PUSH R5 ON STACK
031454  012746  020200                 MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
031460  016605  000020                 MOV     20(SP),R5       ;;GET THE INPUT NUMBER
031464  100004                         BPL     1$              ;;BR IF INPUT IS POS.
031466  005405                         NEG     R5              ;;MAKE THE BINARY NUMBER POS.
031470  112766  000055  000001         MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
031476  005000                 1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
031500  012703  031656                 MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
031504  112723  000040                 MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
031510  005002                 2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
031512  016001  031646                 MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
031516  160105                 3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
031520  002402                         BLT     4$              ;;BR IF DONE
031522  005202                         INC     R2              ;;INCREASE THE BCD DIGIT BY 1
031524  000774                         BR      3$
031526  060105                 4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
031530  005702                         TST     R2              ;;CHECK IF BCD DIGIT=0
031532  001002                         BNE     5$              ;;FALL THROUGH IF 0
031534  105716                         TSTB    (SP)            ;;STILL DOING LEADING 0'S?
031536  100407                         BMI     7$              ;;BR IF YES
031540  106316                 5$:     ASLB    (SP)            ;;MSD?
031542  103003                         BCC     6$              ;;BR IF NO
031544  116663  000001  177777         MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
031552  052702  000060         6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
031556  052702  000040         7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
031562  110223                         MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
031564  005720                         TST     (R0)+           ;;JUST INCREMENTING
031566  020027  000010                 CMP     R0,#10          ;;CHECK THE TABLE INDEX
031572  002746                         BLT     2$              ;;GO DO THE NEXT DIGIT
031574  003002                         BGT     8$              ;;GO TO EXIT
031576  010502                         MOV     R5,R2           ;;GET THE LSD
031600  000764                         BR      6$              ;;GO CHANGE TO ASCII
031602  105726                 8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
031604  100003                         BPL     9$              ;;BR IF NO
031606  116663  177777  177776         MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
031614  105013                 9$:     CLRB    (R3)            ;;SET THE TERMINATOR
031616  012605                         MOV     (SP)+,R5        ;;POP STACK INTO R5
031620  012603                         MOV     (SP)+,R3        ;;POP STACK INTO R3
031622  012602                         MOV     (SP)+,R2        ;;POP STACK INTO R2
```

```
5214  031624  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
5215  031626  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
5216  031630  104401  031656                 TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
5217  031634  016666  000002  000004         MOV     2(SP),4(SP)     ;;ADJUST THE STACK
5218  031642  012616                          MOV     (SP)+,(SP)
5219  031644  000002                          RTI                     ;;RETURN TO USER
5220  031646  023420                $DTBL:    10000
5221  031650  001750                          1000.
5222  031652  000144                          100.
5223  031654  000012                          10.
5224  031656  000004                $DBLK:    .BLKW   4
5225                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE
5226
5227                                ;;**********************************************************************
5228                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
5229                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
5230                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
5231                                ;*CALL:
5232                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5233                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
5234                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
5235                                ;*      .BYTE   M               ;;M=1 OR 0
5236                                ;*                                      ;;1=TYPE LEADING ZEROS
5237                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
5238                                ;*
5239                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
5240                                ;*$TYPOS OR $TYPOC
5241                                ;*CALL:
5242                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5243                                ;*      TYPON                   ;;CALL FOR TYPEOUT
5244                                ;*
5245                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
5246                                ;*CALL:
5247                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
5248                                ;*      TYPOC                   ;;CALL FOR TYPEOUT
5249
5250  031666  017646  000000        $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
5251  031672  116637  000001  032111        MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
5252  031700  112637  032113                MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
5253  031704  062716  000002                ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
5254  031710  000406                        BR      $TYPON
5255  031712  112737  000001  032111  $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
5256  031720  112737  000006  032113        MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
5257  031726  112737  000005  032110  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
5258  031734  010346                        MOV     R3,-(SP)        ;;SAVE R3
5259  031736  010446                        MOV     R4,-(SP)        ;;SAVE R4
5260  031740  010546                        MOV     R5,-(SP)        ;;SAVE R5
5261  031742  113704  032113                MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
5262  031746  005404                        NEG     R4
5263  031750  062704  000006                ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
5264  031754  110437  032112                MOVB    R4,$OMODE       ;;SAVE IT FOR USE
5265  031760  113704  032111                MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
5266  031764  016605  000012                MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
5267  031770  005003                        CLR     R3              ;;CLEAR THE OUTPUT WORD
5268  031772  006105                1$:     ROL     R5              ;;ROTATE MSB INTO "C"
5269  031774  000404                        BR      3$              ;;GO DO MSB
```

```
5270  031776  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
5271  032000  006105                          ROL     R5
5272  032002  006105                          ROL     R5
5273  032004  010503                          MOV     R5,R3
5274  032006  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
5275  032010  105337  032112                  DECB    $OMODE          ;;TYPE THIS DIGIT?
5276  032014  100016                          BPL     7$              ;;BR IF NO
5277  032016  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
5278  032022  001002                          BNE     4$              ;;TEST FOR 0
5279  032024  005704                          TST     R4              ;;SUPPRESS THIS 0?
5280  032026  001403                          BEQ     5$              ;;BR IF YES
5281  032030  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
5282  032032  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
5283  032036  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
5284  032042  110337  032106                  MOVB    R3,8$           ;;SAVE FOR TYPING
5285  032046  104401  032106                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
5286  032052  105337  032110          7$:     DECB    $OCNT           ;;COUNT BY 1
5287  032056  003347                          BGT     2$              ;;BR IF MORE TO DO
5288  032060  002402                          BLT     6$              ;;BR IF DONE
5289  032062  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
5290  032064  000744                          BR      2$              ;;GO DO THE LAST DIGIT
5291  032066  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
5292  032070  012604                          MOV     (SP)+,R4        ;;RESTORE R4
5293  032072  012603                          MOV     (SP)+,R3        ;;RESTORE R3
5294  032074  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
5295  032102  012616                          MOV     (SP)+,(SP)
5296  032104  000002                          RTI                     ;;RETURN
5297  032106     000                  8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
5298  032107     000                          .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
5299  032110     000                  $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
5300  032111     000                  $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
5301  032112  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.

032114  010046                  $TRAP:  MOV     R0,-(SP)        ;;SAVE R0
032116  016600  000002                  MOV     2(SP),R0        ;;GET TRAP ADDRESS
032122  005740                          TST     -(R0)           ;;BACKUP BY 2
032124  111000                          MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
032126  006300                          ASL     R0              ;;POSITION FOR INDEXING
032130  016000  032150                  MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
032134  000200                          RTS     R0              ;;GO TO ROUTINE


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

032136  011646                  $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
032140  016666  000004  000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
032146  000002                          RTI                     ;;RESTORE THE PSW

                                .SBTTL  TRAP TABLE

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

                                ;       ROUTINE
                                ;       -------
032150  032136                  $TRPAD: .WORD   $TRAP2
032152  027412                          $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
032154  031712                          $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
032156  031666                          $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
032160  031726                          $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
032162  031442                          $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)

032164  027744                          $GTSWR  ;;CALL=GTSWR    TRAP+6(104406)  GET SOFT-SWR SETTING

032166  027674                          $CKSWR  ;;CALL=CKSWR    TRAP+7(104407)  TEST FOR CHANGE IN SOFT-SWR
032170  030216                          $RDCHR  ;;CALL=RDCHR    TRAP+10(104410) TTY TYPEIN CHARACTER ROUTINE
032172  030336                          $RDLIN  ;;CALL=RDLIN    TRAP+11(104411) TTY TYPEIN STRING ROUTINE
032174  013126                          GT$DEV  ;;CALL=GTDEVM   TRAP+12(104412) ^C WILL OPEN UP $DEVM

032176  000040                          .BLKW   32.             ;STACK FOR "EIS-FIS" TESTS

032276                          LSTAD:
                                        ;MAX ADDR FOR 8K WITH APT = 37400
                                        ;MAX ADDR FOR 8K NO APT BUT TO SAVE ABS LOADER = 37500
003574                          .END    START
```

```
ABASE = 176500            772#   1253    1294
ACDW1 = 000000           1253
ACDW2 = 000000           1253
ACOM    021530           2364    3111    4734#
ACPUOP= 000000           1253    1268
ADDR    005052           1889*   1907*   1912*   1940*   1977#   4734
ADDW0 = 000000           1253
ADDW1 = 000000           1253
ADDW10= 000000           1253
ADDW11= 000000           1253
ADDW12= 000000           1253
ADDW13= 000000           1253
ADDW14= 000000           1253
ADDW15= 000000           1253
ADDW2 = 000000           1253
ADDW3 = 000000           1253
ADDW4 = 000000           1253
ADDW5 = 000000           1253
ADDW6 = 000000           1253
ADDW7 = 000000           1253
ADDW8 = 000000           1253
ADDW9 = 000000           1253
ADEVCT= 000000           1253    1259
ADEVM = 000017            772#   1253    1295
AENV  = 000000           1253    1264
AENVM = 000000           1253    1265
AFATAL= 000000           1253    1256
AMADR1= 000000           1253    1281
AMADR2= 000000           1253    1285
AMADR3= 000000           1253    1288
AMADR4= 000000           1253    1291
AMAMS1= 000000           1253    1275
AMAMS2= 000000           1253    1283
AMAMS3= 000000           1253    1286
AMAMS4= 000000           1253    1289
AMBAUD  003524           1671#   2334    3132
AMOD  = 030000            986#   2338    2340    3133
AMRB  = 176612            945#   2350    3691
AMRC  = 176610            944#   1805*   1806*   1837*   2342*   2351*   2377*   2378*   3136*   3137*
AMRSRV  014654           2346    3691#
AMRVEC= 000330            918#   2345
AMSGAD= 000000           1253    1261
AMSGLG= 000000           1253    1262
AMSGTY= 000000           1253    1255
AMTST   006562           2268    2273    2332#
AMTYP1= 000000           1253    1276
AMTYP2= 000000           1253    1284
AMTYP3= 000000           1253    1287
AMTYP4= 000000           1253    1290
AMXB  = 176616            947#   3688*
AMXC  = 176614            946#   2352*   3687*   3703*
AMXSRV  014636           2347    3687#
AMXVEC= 000334            919#
ANS1    005724           2169*   2174    2183#
ANS2    005726           2170*   2177    2184#
APASS = 000000           1253    1258
```

```
APRIOR= 000000  1253
APTCSU= 000040  4765  5024#
APTENV= 000001  4758  4980  5022#  5055
APTSIZ= 000200  1740  5021#
APTSPO= 000100  4760  4982  5023#
ASHC1   005236  2032  2038#
ASHC2   005306  2054  2060#
ASHC3   005360  2076  2081#
ASHC4   005400  2085  2092#
ASH1    005106  1986  1991#  2182
ASH2    005146  2003  2008#
ASH3    005170  2012  2017#
ASH4    005210  2021  2026#
ASWREG= 000000  1253  1266
ATESTN= 000000  1253  1257
AUNIT = 000000  1253  1260
AUSWR = 000000   772#  1253  1267
AVECT1= 000300   772#  1253  1292
AVECT2= 000000  1253  1293
BITHLD  013714  3425*  3429  3443#
BIT0  = 000001   891#  1058  1154  2384  3187
BIT00 = 000001   88*#   891
BIT01 = 000002   880#   890
BIT02 = 000004   879#   889
BIT03 = 000010   878#   888
BIT04 = 000020   877#   887
BIT05 = 000040   876#   886
BIT06 = 000100   875#   885
BIT07 = 000200   874#   884
BIT08 = 000400   873#   883
BIT09 = 001000   872#   882
BIT1  = 000002   890#  1025  1045  1081  1153  2422
BIT10 = 002000   871#  1038  1072  2266  5043
BIT11 = 004000   870#  1071  1112  2332
BIT12 = 010000   869#  1037  1052  1070  1087  1111  1954  2189  2214  2253  2322  2362  2380
                2409  2447  2485  4074  4090  4111  4454  4470  4492
BIT13 = 020000   868#  1036  1051  1069  2418  5050
BIT14 = 040000   867#  1035  1050  1068  1086  1094  2456
BIT15 = 100000   866#  1034  1049  1067  1085  1110  1129  2223  4200  4389
BIT2  = 000004   889#  1044  1080  1152  2460
BIT3  = 000010   888#  1043  1079  1118  1151  2283  2336  3125
BIT4  = 000020   887#  1078  1117  1132  1133  1150  3149  5132
BIT5  = 000040   886#  1042  1077  1116  1149  2501  3089
BIT6  = 000100   885#  1041  1057  1076  1115  1131  1985
BIT7  = 000200   884#  1040  1056  1075  1114  1130  1148  2199
BIT8  = 000400   883#  1074  1113  2599  2637  2698  2748  2772  2817  2884  2921
BIT9  = 001000   882#  1039  1073  2618  2645  2723  2753  2846
BOTMON  005014  1916  1933  1946  1964#
BPTVEC= 000014   898#
BREAK = 000001  1058#
BUFTST  021604  3105  4734#
B110  = 001000   994#
B1200 = 003400   999#
B134  = 001400   995#
B150  = 002000   996#
B1800 = 004000  1000#
```

```
B19200= 007400           1007#
B2000 = 004400           1001#
B2400 = 005000           1002#   1672    1673    1674
B300  = 002400            997#
B3600 = 005400           1003#
B4800 = 006000           1004#
B50   = 000000            992#
B600  = 003000            998#
B7200 = 006400           1005#
B75   = 000400            993#
B9600 = 007000           1006#   1669    1670    1671
CDET  = 010000           1037#   1070#
CHAR5 = 000000           1018#
CHAR6 = 000004           1019#
CHAR7 = 000010           1020#
CHAR8 = 000014           1021#   2229    2338    2340    2386    2388    2424    2426    2462    2464    3133
CHKCON  012624           1952    2187    2212    2251    2319    2360    2407    2445    2483    2968    3167#   4072    4088
                         4109    4452    4468    4490
CHR5  = 000000           1098#
CHR6  = 002000           1099#
CHR7  = 004000           1100#
CHR8  = 006000           1101#   2286
CKSWR = 104407           3254    5039    5065    5341#
CKOBAD  014006           2711    3490#
CK1BAD  014120           2736    3538#
CLK   = 177546            939#   1831*   2204*
CLKTST  005760           2199#
CLKVEC= 000100            916#   1693*   1694*   2201*   2202*
CLOCK   021676           2216    3107    4734#
CLOPT   021442           3076    4734#
CLPRES  012614           2371    3072*   3075*   3155#
CLRBAD  013716           1845    3450#
CLT1    021462           2411    3112    4734#
CLT2    021473           2449    3113    4734#
CLT3    021504           2487    3114    4734#
CL1TST  006744           2333    2358    2361    2363    2371#
CL2TST  007156           2381    2405    2408    2410    2418#
CL3TST  007332           2419    2443    2446    2448    2456#
COMCHR  014732           2309*   2314    2349*   2355    3688    3692    3696    3698*   3701*   3706#   3714    3715    3738
                         3742    3744*   3747*   4734
COMHLD  014734           2298*   2299    2350*   3691*   3692    3707#   3735*   3736    3738    4734
COMPAT  023216           3002    4734#
COMP1   003510           1658#   1680*   1684*   1688*   1696*   1800    1848    2653    2994    3000    3004*   3182    4196
                         4385
COMP2   003512           1659#   1683*   1689*   1697*   1802    1850    2605    2624    2758    2996    3005*   3184    4198
                         4387
CONSOL= 010000            981#
COPDUN  023124           4639    4734#
COPFLG  003506           1657#   1687*   1698*   1774    1856    4567*
COPMSG  023023           4568    4734#
COPY    020714           1858    4567#   4641
CR    = 000015            806#   2236    3581    3594    4804    4814
CRC   = 000010           1151#   3939    4328
CRLF  = 000200            807#   1762    4734    4775    4814
CTHLD   013422           3334*   3338    3352#
CTREG   013516           3362*   3366*   3378#   3391*   3395*
```

```
CTREG1  013520          3361* 3369* 3379# 3390* 3398*
CTS   = 020000          1036# 1069#
CT1   = 000000           982# 2386  2388
CT2   = 050000           983# 2424  2426
CT3   = 060000           984# 2462  2464
C1BAUD  003526          1672# 2382
C2BAUD  003530          1673# 2420
C3BAUD  003532          1674# 2458
DATCHK  021406          4623  4714#
DD    = 000040          1149# 3963
DDISP = 177570           813# 1236  1728
DEFBD   021746          1813  4734#
DEVM    023313          3261  4734#
DF      027364          1315  1318  1321  1324  1327  1330  1333  1336  1339  1342  1345  1348  1351
                        1390  1393  1396  1399  1402  1405  1408  1411  1414  1417  1420  1423  1426
                        1429  1432  1435  1438  1444  1447  1450  1453  1456  1459  1462  1465  1492
                        1495  1498  1501  1507  1510  1513  1516  1549  1552  1555  1558  1561  1564
                        1567  1570  1573  1576  1584  1587  1593  1596  4734#
DF1     027372          1354  1357  1360  1363  1372  1375  1378  1381  1468  1471  1474  1477  1480
                        1483  1486  1489  1531  1540  4734#
DF2     027402          1366  1369  1384  1387  1441  1534  1543  4734#
DH1     026441          1330  1390  1396  1399  1402  1405  1408  1417  1420  1423  1584  1587  4734#
DH2     026460          1315  1318  1324  4734#
DH3     026504          1321  1327  4734#
DH4     026546          1333  1393  4734#
DH5     026577          1336  1339  1342  1345  1348  1498  1501  4734#
DH6     026634          1351  1411  1414  1426  1429  1432  1435  1438  1444  1447  1450  1453  1456
                        1459  1462  1465  1492  1495  1507  1510  1513  1516  1549  1552  1555  1558
                        1561  1564  1567  1570  1573  1576  1593  1596  4734#
DH7     026701          1354  1357  1360  1363  1372  1375  1378  1381  1468  1471  1474  1477  1480
                        1483  1486  1489  1531  1540  4734#
DH8     026770          1366  1369  1384  1387  1441  1534  1543  4734#
DISPLA  001142          1236# 1728* 1736* 5042*
DISPRE  000174          1163# 1736
DIVO    005476          2117  2124#
DNA   = 100000          1110#
DNAIE = 000040          1116#
DONE  = 000200          1040# 1075# 1130# 2294  2506  2514  2533  2546  4545  4685  4704
DROP    022446          3102  4734#
DRV0    021572          3108  4076  4092  4113  4734#
DRV1    021577          3109  4456  4472  4494  4734#
DSC   = 100000          1067#
DSI   = 100000          1034#
DSIE  = 000040          1042# 1077#
DSRDY = 001000          1039# 1073#
DSWR  = 177570           812# 1235  1727
DTR   = 000002          1045# 1081# 1805  1806  1837  2287  2342  2377  2378  3136  3137
DT1     027062          1330  1390  1396  1399  1402  1405  1408  1417  1420  1423  1584  1587  4734#
DT10    027212          1354  1357  1360  1363  1468  1471  1474  1477  1531  4734#
DT11    027234          1366  1369  1441  1534  4734#
DT12    027256          1372  1375  1378  1381  1480  1483  1486  1489  1540  4734#
DT13    027300          1384  1387  1543  4734#
DT14    027322          1411  1414  1426  1429  1432  1435  1450  1456  1495  1507  1510  1513  1516
                        1549  1552  1555  1558  1561  1564  1567  1570  1573  1576  1596  4734#
DT15    027340          1498  4734#
DT16    027352          1501  4734#
```

```
DT2     027070          1315    1318    1324    4734#
DT3     027100          1321    1327    4734#
DT4     027114          1333    1393    4734#
DT5     027124          1336    4734#
DT6     027136          1339    4734#
DT7     027150          1342    4734#
DT8     027162          1345    1348    4734#
DT9     027174          1351    1438    1444    1447    1453    1459    1462    1465    1492    1593    4734#
DUN     022574          1957    2192    2325    4080    4094    4115    4460    4474    4496    4734#
DXTST   003476          1644#   2499*   2584*   2685*   2775*   2820*   2849*   2887*   2925*   3829    3961    4030    4053
                        4123    4125    4501    4734
DXTST0  007506          2373    2457    2481    2484    2486    2499#
DXTST1  007750          1854    2502    2563    2584#
DXTST2  010404          2658    2668    2685#
DXTST3  010754          2759    2772#
DXTST4  011130          2773    2786    2817#
DXTST5  011410          2847    2856    2884#
DXTST6  011526          2885    2898    2907    2921#
DX1   = 000020          1132#   2631    2742    2852    2868    4150    4180    4203    4223    4247    4279    4292    4312
                        4325    4365    4476    4497    4596    4609
DOBAD   003004          1614#   3450    3452    3468    3475    3497    3498
DOBUF   002604          1613#   2538    2551    2561    3980    4002    4593    4715
DOCDAT  016174          3981    3996#
DOCERR  002574          1609#   2591*   2691*   2777*   2797*   2837*   2889*   4028    4043    4048*   4052*   4734
DOCMER  002572          1608#   3881*   3996*   4007*   4021    4067*   4104*
DOCNT   002562          1604#   2692*   2750    4106*   4107
DOCRC   002602          1612#   3941*   3945*   4023    4038    4049*
DODISP  015226          2607*   2610*   2714*   2779*   2801*   2821*   2839*   2891*   2900*   2930*   3783    3784#   3805*
                        3828*   3856*   3888*   3902*   3922*   3935*   3981*   4136*   4145*   4157*   4165*
DODUN   017052          4056    4127    4136    4145    4157    4165    4175#
DOEBUF  016136          2801    3967    3979#
DOERR   002564          1605#   2590*   2690*   2776*   2796*   2836*   2888*   3846    3863    3874*   3880*   3884*   3916
                        3918*   3949    3960*   4066*   4103*   4734
DOEXP   002576          1610#   4005*   4012*   4734
DOFBUF  015244          2610    2891    3800#   3877    3888
DOINIT  016736          2779    4134#
DOINV   017034          2930    4165#
DONEXT  016714          3935    3947    4050    4123#
DONXT1  016442          4066#   4124
DONXT2  016622          4103#   4126
DOPAT   002560          1603#   2589*   2639    4095*
DORCHK  015666          3902    3914#
DOREC   002600          1611#   4006*   4734
DOREST  016762          2821    4145#
DORSEC  015632          2607    2714    2839    3882    3897#   3920    3922    4055    4070
DOSEC   002570          1607#   2592*   2709*   2799*   2834*   2927*   2953*   3495    3804    3826    3876    3879*   3899
                        4001    4069    4085*   4734
DOSTAT  017016          2900    4157#
DOTMO   013522          2641    3360*   3371*   3380#
DOTRK   002566          1606#   2586*   2662*   2669*   2704*   2798*   2833*   2926*   2952*   3493    3802    3824    3897
                        3999    4078    4082    4084*   4734
DOWCHK  015402          3828    3844#
DOWSEC  015326          3805    3824#
D1BAD   003254          1632#   3455    3457    3516    3523    3545    3546
D1BUF   003054          1631#   4363    4391    4621    4716
D1CDAT  020010          4364    4378#
```

```
D1CERR  003044  1627#  2596*  2695*  2865*  4416  4425  4431*  4435*  4734
D1CMER  003042  1626#  4271*  4378*  4396*  4410  4447*  4485*
D1CNT   003032  1622#  2696*  2755  4487*  4488
D1CRC   003052  1630#  4330*  4334*  4412  4421  4432*
D1DISP  015234  2626*  2629*  2739*  2850*  2867*  3786  3787#  4149*  4202*  4222*  4246*  4278*  4291*
                4311*  4324*  4364*
D1DUN   017066  4149  4179#
D1EBUF  017752  4350  4362#
D1ERR   003034  1623#  2595*  2694*  2864*  4236  4253  4264*  4270*  4274*  4305  4307*  4338  4349*
                4446*  4484*  4734
D1EXP   003046  1628#  4394*  4401*  4734
D1FBUF  017102  2629  4190#  4267  4278
D1INT   015230  3781  3786#
D1NEXT  020520  4324  4336  4433  4501#
D1NXT1  020246  4446#  4502
D1NXT2  020426  4484#  4503
D1PAT   003030  1621#  2594*  2647  4475*
D1RCHK  017524  4291  4303#
D1REC   003050  1629#  4395*  4734
D1REST  017000  2850  4149#
D1RSEC  017470  2626  2739  2867  4272  4286#  4309  4311  4436  4450
D1SEC   003040  1625#  2597*  2734*  2863*  3543  4194  4220  4266  4269*  4288  4383  4449  4465*
                4734
D1TMO   013616  2649  3389*  3400*  3407#
D1TRK   003036  1624#  2587*  2663*  2670*  2729*  2862*  3541  4192  4218  4286  4381  4458  4462
                4464*  4734
D1WCHK  017240  4222  4234#
D1WSEC  017204  4202  4218#
EBHLD   007742  2552*  2556*  2557  2567#  4734
EBUF  = 000003  1138#  2539  3982  4365  4698
EBUFF   021344  4592  4620  4696#
EIPRES  012612  1983  3046*  3052*  3154#
EISCNT  005730  1982*  2180*  2185#
EISDON  005732  2181  2187#
EISFIS  005056  1953  1955  1958  1982#
EISTXT  021705  3054  3106  4734#
EMTVEC= 000030  901#  1719*  1720*
EM1     023324  1315  1318  1324  4734#
EM10    023637  1351  4734#
EM11    023664  1354  4734#
EM12    023716  1357  4734#
EM13    023750  1360  1507  4734#
EM14    024001  1363  4734#
EM15    024032  1366  4734#
EM16    024066  1369  4734#
EM17    024112  1372  1510  4734#
EM18    024144  1375  4734#
EM19    024176  1378  1516  4734#
EM2     023343  1321  1327  4734#
EM20    024227  1381  4734#
EM21    024260  1384  4734#
EM22    024314  1387  4734#
EM23    024340  1390  4734#
EM24    024351  1393  4734#
EM25    024366  1396  4734#
EM26    024403  1399  4734#
```

```
EM27    024420          1402    4734#
EM28    024435          1405    4734#
EM29    024454          1408    4734#
EM3     023365          1330    4734#
EM30    024474          1411    1426    1429    1432    1435    1450    1456    1558    1564    4734#
EM31    024505          1414    1561    1567    1570    4734#
EM32    024516          1417    4734#
EM33    024561          1420    4734#
EM34    024622          1423    4734#
EM35    024661          1438    4734#
EM36    024726          1441    4734#
EM37    024775          1444    4734#
EM38    025021          1447    4734#
EM39    025067          1453    4734#
EM4     023431          1333    4734#
EM40    025117          1459    4734#
EM41    025162          1462    4734#
EM42    025216          1465    1593    4734#
EM43    025237          1468    4734#
EM44    025301          1471    4734#
EM45    025341          1474    4734#
EM46    025402          1477    4734#
EM47    025441          1480    4734#
EM48    025503          1483    4734#
EM49    025543          1486    4734#
EM5     023454          1336    4734#
EM50    025604          1489    4734#
EM51    025643          1492    4734#
EM52    025667          1495    1549    1552    1555    1573    1576    1596    4734#
EM53    025713          1498    1501    4734#
EM54    025762          1513    4734#
EM55    025777          1522    4734#
EM56    026034          1525    4734#
EM57    026071          1531    4734#
EM58    026126          1534    4734#
EM59    026167          1537    4734#
EM6     023502          1339    4734#
EM60    026225          1540    4734#
EM61    026262          1543    4734#
EM62    026323          1546    4734#
EM63    026361          1584    4734#
EM64    026416          1587    4734#
EM7     023530          1342    4734#
EM8     023556          1345    4734#
EM9     023607          1348    4734#
ENDPAS  022633          2970    4734#
ENVPAR= 001400          1105#
EOP     011706          2565    2655    2760    2937    2966#
EPAR  = 000060          1013#
ERFLG   021226          4573*   4584*   4585    4591*   4600*   4601    4607*   4613*   4614    4645#
ERR   = 100000          1049#   1085#   1129#
ERRNUM  031430          4734    5127*   5152#
ERRVEC= 000004           894#   1725    1726*   1737*   1873*   1874*   1949*   1950*   3016*   3017*   3079*   3080*
EXAMKB  013104          1871    3250#   3340    3365    3394    3431
EXP0    007744          2568#   4734
EXP1    007746          2569#   4734
```

```
FBUF   = 000001          1137#  2525   3806   4203
FIN      003500          1652#  2611*  2630*  2716*  2741*  2781*  2803*  2822*  2838*  2851*  2866*  2893*  2902*
                         2932*  3363   3392   4081*  4105*  4175*  4179*  4461*  4486*
FIS      005542          2136   2146#
FISEIS   022542          2191   4734#
FLGHLD   013420          3333*  3338   3351#
FRERR  = 020000          1051#
GETSEC   021234          4571   4653#
GNS    = ****** U        1761   5333   5334   5335   5336   5337   5339   5341   5342   5343   5344
GTDEVM=  104412          1783   5344#
GTSWR  = 104406          1756   1766   5339#
GT$DEV   013126          3261#  3303   4855   4941   5344
HFDUP  = 000010          1118#
HILIM    013076          2702*  2707*  2727*  2732*  3198*  3237   3244#
HINUM    013100          3216   3224   3229*  3246#
HLD42    005054          1706*  1964   1978#
HOLD     003534          1675#  2228*  2229*  2230   2277*  2278*  2279   2334*  2338*  2340*  2341   2382*  2386*
                         2388*  2390   2420*  2424*  2426*  2428   2458*  2462*  2464*  2466   3132*  3133*  3134
HT     = 000011           804#  4773   4814
ID     = 000001          1154#  2792
IE     = 000100          1041#  1057#  1076#  1115#  1131#  1831   2204   2238   2249   2310   2311   2351   2352
                         2398   2399   2436   2437   2474   2475   2612   2631   2717   2742   2782   2804   2823
                         2840   2852   2868   2894   2903   2933   3607   3623   3634   3650   3660   3676   3687
                         3703   3713   3721   3748   3752   3806   3831   3833   3857   3889   3903   3923   3936
                         3982   4096   4116   4137   4146   4150   4158   4166   4176   4180   4203   4223   4247
                         4279   4292   4312   4325   4365   4476   4497   4539
INITAL=  000011          1141#  2510   4137
INTFLG   014202          1885*  1887   1903*  1905   2203*  2207   3024*  3028   3047*  3050   3062*  3065   3565*
                         3567#
INTSRV   014174          1873   2201   3016   3018   3565#
INVADR=  000002          1153#  2939
INVCT    011704          2924*  2948   2951*  2959#
IOTVEC=  000020           899#
LDATA  = 000004          1152#
LDOBAD   013750          3466#  3848   3926
LD1BAD   014062          3514#  4238   4315
LF     = 000012           805#  3583   3590   4734   4808   4814
LINCT    014332          2235*  2242   3595*  3600#
LOLIM    013074          2701*  2706*  2726*  2731*  3197*  3235   3243#
LONUM    013102          3215   3222   3228*  3247#
LOOP     004364          1775   1801   1803   1815   1819   1827#  3006
LSTAD    032276          1892   1901   1929   1937   2146   2171   4653   5348#
LSTCHR   015024          2275*  3718   3720*  3729#
MAINT  = 000002          1025#  2338   2386   2424   2462
MAXMEM   012622          1876   3037*  3159#
MAXSEC   021222          4627   4643#  4666*
MAXSK    003504          1655#  1769*  2750   2755   2998*  4107   4488
MAXTRK   003502          1653#  1768*  1853*  2660   2664*  2667   2671*  2688*  2702   2727   2993*  4082   4462
MCLK   = 004000          1112#  2281   2285
MEM      021725          3041   4734#
MEMHLD   005046          1909*  1910   1942*  1975#  4734
MEMORY   022561          1956   4734#
MEMTST   004502          1857   1871#
MM     = 010000          1111#  2285
MR     = 000400          1113#  2280   2321
MSG1     022650          2974   4734#
```

```
MSG2    022677            2978    4734#
MSG3    022733            2985    4734#
MSG4    022767            2989    4734#
MULO    005436            2101    2107#
MYTYPE  031320            5052    5127#
NOCON   021631            3151    4734#
NOEIS   005760            1987    2188    2190    2193#
NOTPRE  022133            3055    3077    3129    4734#
NULL    012102            2992    3008#   5145
NXTSEC  020532            3875    4068    4265    4448    4512#
ODDPAR= 001000            1106#   2286
OK      022500            4734#
OPAR  = 000020            1014#   2338    2340    3133
ORERR = 040000            1050#   1086#
PARAM = 177420             937#   2230*   2279*   2341*   2390*   2428*   2466*   3134*   5134*   5136*
PARERR= 010000            1052#   1087#
PASS    022602            4734#   5138
PAT     005050            1900*   1904    1910    1922*   1941*   1976#   4734
PATFLG  012752            3196*   3204*   3207#   3231
PATT    022511            4093    4473    4734#
PERR    031436            1842*   2986    5149*   5156#
PIRQ  = 177772             811#
PIRQVE= 000240             905#
PORT    021555            2258    4734#
PRB   = 177516             942#   3580*
PRBAUD  003520            1669#   2228
PRCHR   014330            2236*   3580    3581    3583    3585    3587*   3590*   3592*   3594*   3599#
PRINT   021544            2255    3115    3128    4734#
PRPORT  012620            2247    2256    3121*   3140*   3157#
PRPRES  012616            2225    3120*   3141*   3156#
PRS   = 177514             941#   2238*   2249*   3122    3138    3574
PRSRV   014204            2232    3574#
PRT   = 020000             985#   2229
PRTST   006060            2200    2210    2213    2215    2223#
PRVEC = 000200             914#   2232*   2233*
PR0   = 000000             828#   1835
PR1   = 000040             829#
PR2   = 000100             830#
PR3   = 000140             831#
PR4   = 000200             832#   1705
PR5   = 000240             833#
PR6   = 000300             834#
PR7   = 000340             835#
PS    = 177776             808#    809
PSERR   031440            1843*   2990    3866*   3952*   4046*   4256*   4341*   4428*   5157#
PSW   = 177776             809#
PWRVEC= 000024             900#
RACT  = 004000            1071#
RAND    012754            2703    2708    2728    2733    3199    3213#
RANDOM  022524            4114    4495    4734#
RDCHR = 104410            4935    5342#
RDLIN = 104411            1794    5343#
RDWR    021304            4579    4595    4608    4678#
RDY   = 000200            1056#   1114#   1148#
REGHLD  013712            3424*   3429    3442#
RESTOR= 000017            1144#   3857    3889    3923    3936    4146    4150    4247    4279    4312    4325
```

```
RESVEC= 000010   895#  3018*  3019*  3081*  3082*
RING  = 040000  1035#  1068#
RNF   = 000020  1150#  3849   3885   3919   3927   4239   4275   4308   4316   4549
RSEC  = 000007  1140#  3903   4292   4542   4580   4609
RSTAT = 000013  1142#  4158
RTS   = 000004  1044#  1080#  1805   1806   1837   2287   2342   2377   2378   3136   3137
RUN     022467  2217   2260   2365   2412   2450   2488   4734#
RXCS  = 177170   955#  2505   2510*  2513   2518   2525*  2532   2539*  2545   2612*  2631*  2717*  2742*
                2782*  2788   2804*  2823*  2829   2840*  2852*  2858   2868*  2894*  2903*  2909   2933*
                2944   3765   3767   3773   3776   3780   3806*  3831*  3833*  3844   3857*  3889*  3903*
                3914   3923*  3936*  3982*  4096*  4116*  4137*  4146*  4150*  4158*  4166*  4176*  4180*
                4203*  4223*  4234   4247*  4279*  4292*  4303   4312*  4325*  4365*  4476*  4497*  4539*
                4542*  4544   4576   4582   4598   4611   4682*  4684   4698*  4703   4734
RXDB  = 177172   956#  2526*  2540   3807*  3810*  3983   4204*  4207*  4366   4699
RXES  = 177172   957#  2792   2939   3768   3777   3963   4549   4734
RXSA  = 177174   958#  2928*  2954*  3471   3519   3769   3778   3827*  3900*  4134*  4221*  4289*  4540*
                4541*  4551   4553*  4681*  4734
RXSRV   015126  2602   2621   3765#
RXVEC = 000264   924#  2602*  2603*  2621*  2622*
SCOM    021515  2324   3110   4734#
SECCNT  021224  4625*  4627   4644#  4669*
SECT    021232  4575*  4630   4632*  4636*  4647#  4680
SEKFLG  003516  1662#  3852   3854*  3867   3869*  3930   3932*  3953   3955*  4242   4244*  4257   4259*
                4319   4321*  4342   4344*  4538*  4556*
SEKTST  020612  3851   3887   3921   3929   4241   4277   4310   4318   4538#
SEND  = 000020  1117#  2289   3725
SERIAL  022614  1790   2982   4734#  5143
SETPAT  012644  1846   3179#
SETVEC  013314  2304   2344   2391   2429   2467   3314#
SIZE    012106  1812   3016#
SMBAUD  003522  1670#  2277
SMFLG   003514  1661#  2276*  2326*  3252
SMOD  = 030000   987#  2278
SMPAR = 176622   951#  2286*
SMRB  = 176622   950#  2298   3735
SMRC  = 176620   949#  2287*  2293   2310*  3748*
SMRSRV  015026  2306   3735#
SMRVEC= 000340   921#  2305
SMIST   006246  2224   2226   2245   2252   2254   2259   2266#
SMXB  = 176626   953#  2290*  3714*
SMXC  = 176624   952#  2280*  2281*  2285*  2289*  2311*  2321*  3713*  3721*  3725*  3752*
SMXSRV  014736  2307   3713#
SMXVEC= 000344   922#
SPRS    014334  3574*  3575   3601#  4734
SREC  = 002000  1038#  1072#
SRSYN = 000020  1078#  2287
SRXCS   015236  3767*  3776*  3790#  4734
SRXES   015240  3768*  3777*  3791#  3849   3885   3919   3927   3939   4239   4275   4308   4316   4328
                4734
SRXSA   015242  3769*  3778*  3792#  4734   5128
STACK = 001100   799#  1700   1717
START   003574  1171   1174   1693#  5351
START1  003536  1179   1680#
START2  003544  1181   1683#
START3  003556  1177   1687#
STKLMT= 177774   810#
```

```
STSYN = 000400           1074#
SWR     001140           1235#  1715   1727*  1729   1735*  1742*  1754   1954   2189   2214   2253   2322   2362
                         2409   2447   2485   4074   4090   4111   4454   4470   4492   4824   4869*  5043   5050
                         5062
SWREG   000176           1164#  1735   1754   4824   4837
SW0   = 000001            863#
SW00  = 000001            853#   863
SW01  = 000002            852#   862
SW02  = 000004            851#   861
SW03  = 000010            850#   860
SW04  = 000020            849#   859
SW05  = 000040            848#   858
SW06  = 000100            847#   857
SW07  = 000200            846#   856
SW08  = 000400            845#   855
SW09  = 001000            844#   854
SW1   = 000002            862#
SW10  = 002000            843#
SW11  = 004000            842#
SW12  = 010000            841#
SW13  = 020000            840#
SW14  = 040000            839#
SW15  = 100000            838#
SW2   = 000004            861#
SW3   = 000010            860#
SW4   = 000020            859#
SW5   = 000040            858#
SW6   = 000100            857#
SW7   = 000200            856#
SW8   = 000400            855#
SW9   = 001000            854#
SXMIT = 000010           1043#  1079#
SYNC1 = 040000           1094#
SYNC2 = 000000           1093#
S1      003614           1681   1697#
S2      003620           1685   1698#
S3      003624           1690   1700#
TBITVE= 000014            896#
TERR    031432           1771*  2975   5148*  5154#
TIMDO   013424           2614   2719   2784   2806   2825   2842   2896   2905   2935   3360#
TIMD1   013524           2633   2744   2854   2870   3389#
TIMER   013340           2206   2241   2313   2354   2401   2439   2477   3333#
TIMER1  013620           1829   1840   3167   3413#
TIMER2  013632           2292   2504   2512   2531   2544   3424#  4543   4683   4702
TKVEC = 000060            903#
TK1B  = 176502            961#   2397   3611
TK1S  = 176500            960#   2398*  3063
TK1SRV  014354           2393   3611#
TK1VEC= 000300            926#   2392
TK2B  = 176512            966#   2435   3638
TK2S  = 176510            965#   2436*
TK2SRV  014454           2431   3638#
TK2VEC= 000310            928#   2430
TK3B  = 176522            971#   2473   3664
TK3S  = 176520            970#   2474*  3067
TK3SRV  014554           2469   3664#
```

```
TK3VEC= 000320          930#  2468
TPVEC  = 000064          904#
TP1B   = 176506          963#  3608*
TP1S   = 176504          962#  2399*  3607*  3623*
TP1SRV  014336          2394   3607#
TP1VEC= 000304           927#
TP2B   = 176516          968#  3635*
TP2S   = 176514          967#  2437*  3634*  3650*
TP2SRV  014436          2432   3634#
TP2VEC= 000314           929#
TP3B   = 176526          973#  3661*
TP3S   = 176524          972#  2475*  3660*  3676*
TP3SRV  014536          2470   3660#
TP3VEC= 000324           931#
TRACK   021230          4574*  4635*  4646#  4678
TRAPVE= 000034           902#  1721*  1722*
TRK     022504          4077   4457   4734#
TRTVEC= 000014           897#
TSERR   031434          1772*  2979   3865*  3951*  4045*  4255*  4340*  4427*  5155#
TYPDS  = 104405         2972   2976   2980   2987   2991   3040   4079   4459   5112   5141   5337#
TYPE   = 104401         1748   1790   1791   1793   1813   1820   1956   1957   2191   2192   2216   2217   2255
                        2258   2260   2324   2325   2364   2365   2411   2412   2449   2450   2487   2488   2970
                        2974   2978   2982   2983   2985   2989   2992   3002   3041   3054   3055   3076   3077
                        3100   3102   3128   3129   3151   3261   3264   3280   3301   4076   4077   4080   4092
                        4093   4094   4113   4114   4115   4456   4457   4460   4472   4473   4474   4494   4495
                        4496   4568   4639   4778   4835   4836   4839   4850   4860   4871   4890   4939   4944
                        4947   4951   5045   5053   5076   5093   5095   5098   5100   5115   5120   5138   5143
                        5144   5145   5216   5285   5333#
TYPOC  = 104402         3263   4838   5084   5108   5334#
TYPON  = 104404         5336#
TYPOS  = 104403         5335#
T1CHR   014432          2396*  2402   3608   3612   3616   3618*  3621*  3626#  4734
T1HLD   014434          2397*  3611*  3612   3627#  4734
T2CHR   014532          2434*  2440   3635   3639   3643   3645*  3648*  3653#  4734
T2HLD   014534          2435*  3638*  3639   3654#  4734
T3CHR   014632          2472*  2478   3661   3665   3669   3671*  3674*  3679#  4734
T3HLD   014634          2473*  3664*  3665   3680#  4734
ULITE  = 040000          988#  5134   5136
USEL   = 000020         1133#  3780
WARNIN  022150          1820   4734#
WCP     003300          1642#  3180   3809   3998   4206   4380
WDDSEC= 000015          1143#  3831
WSEC   = 000005         1139#  3833   4166   4223   4596
$APTHD  001000          1195   1201#
$ASTAT= ****** U        5002   5017
$ATYC   030562          4973   4975#
$ATY1   030536          4971#
$ATY3   030544          4763   4972#
$ATY4   030554          4974#  5058
$AUTOB  001134          1232#  1758*  4832   4967
$BASE   001244          1294#
$BDADR  001122          1227#
$BDDAT  001126          1229#
$BELL   001160          1245#  5045   5068
$CHARC  027670          4780*  4790*  4797   4806*  4811#
$CKSWR  027674          4824#  5341
```

```
$CMTAG  001100          1215#   1712    1713    1723
$CM3  = 000000          1245#
$CNTLC  030474          4850    4939    4960#
$CNTLG  030506          4835    4962#
$CNTLU  030501          4860    4961#
$CPUOP  001216          1268#
$CRLF   001165          1247#   3280    4779    4814    4871    4960    5053    5068    5076    5095    5100    5120
$DBLK   031656          5182    5216    5224#
$DEVCT  001200          1259#   2967*   3343*   3368*   3374*   3397*   3403*   3434*   3596*   3622*   3649*   3675*   3702*
                        3722*   3726*   3749*
$DEVM   001246          1295#   1816    1818    1985    2199    2223    2266    2283    2332    2336    2375    2380    2384
                        2418    2422    2456    2460    2501    2599    2618    2637    2645    2698    2723    2748    2753
                        2772    2817    2846    2884    2921    3091    3125    3149    3262    3278*   5132
$DTBL   031646          5185    5220#
$ENV    001210          1264#   1707    1752    1781    1814    2270    3250    4758    4980    5004    5055
$ENVM   001211          1265#   1740    4760    4765    4982
$ERFLG  001103          1218#   5040*   5068
$ERMAX  001115          1224#
$ERROR  031004          1719    5038#
$ERRPC  001116          1225#   4734    5047*   5048*   5049    5068    5082
$ERRTB  001250          1312#   5090
$ERRTY  031144          5075#   5147
$ERTTL  001112          1222#   5046*   5068
$ETABL  001210          1263#
$ETEND  001250          1207    1296#
$FATAL  001172          1256#   5008*
$FFLG   031002          4971*   4974*   5002    5011*   5019#
$FILLC  001156          1243#   4783    4814
$FILLS  001155          1242#   4814
$GDADR  001120          1226#
$GDDAT  001124          1228#
$GTSWR  027744          4836#   5339
$HD   = 000000           783
$HIBTS  001000          1202#
$ICNT   001104          1219#
$INTAG  001135          1233#   4852    4872    4967
$ITEMB  001114          1223#   5049*   5057    5068    5079    5127
$LF     001166          1248#   4814    4951    4960    5068
$LFLG   031001          5012*   5018#
$LPADR  001106          1220#
$LPERR  001110          1221#
$LSTCN= 177777           772#
$LSTIN= 000000           772#
$LSTST= 177777           772#
$LSTTA= 000000           772#
$MADR1  001222          1281#
$MADR2  001226          1285#
$MADR3  001232          1288#
$MADR4  001236          1291#
$MAIL   001170          1203    1207    1254#   1739    1752    4758    5055
$MAMS1  001220          1275#
$MAMS2  001224          1283#
$MAMS3  001230          1286#
$MAMS4  001234          1289#
$MBADR  001002          1203#
$MFLG   031000          4972*   4978    5013*   5017#
```

```
$MNEW   030524          1793    3264    4839    4965#
$MSGAD  001204          1261#   4988*   4991
$MSGLG  001206          1262#   4993*
$MSGTY  001170          1255#   4986    4994*   5006    5010*
$MSWR   030513          4836    4963#
$MTYP1  001221          1276#
$MTYP2  001225          1284#
$MTYP3  001231          1287#
$MTYP4  001235          1290#
$NULL   001154          1241#   4785    4814
$OCNT   032110          5257*   5286*   5299#
$OMODE  032112          5252*   5256*   5261    5264*   5275*   5301#
$PASS   001176          1258#   1739*   2272    2657    2686    2966*   2971    3187    5139
$PASTM  001006          1205#
$QUES   001164          1246#   3301    4814    4890    4944    4960    5068
$RDCHR  030216          4903#   5342
$RDDEC= ****** U        5344
$RDLIN  030336          4931#   5343
$RDOCT= ****** U        5344
$RDSZ = 000010          4924#
$R2A  = ****** U        5344
$SAVRE= ****** U        5344
$SETUP= 000106          1678#   1718    1719    1721    1723    1748    1749    4819    4967    5039    5065    5067
$STUP = 177777          1678#
$SWR  = 132000           772#    783    1245    1723    5031    5032    5033    5034    5035    5043    5050    5062    5066
                        5068
$SWREG  001212          1266#   1742
$TESTN  001174          1257#   5128*
$TKB    001146          1238#   3271    4817    4828    4845    4907    4913
$TKS    001144          1237#   1701*   3269    4817    4826    4842    4854*   4874*   4905    4911
$TN   = 000001           772#    783
$TPB    001152          1240#   4803*   4814
$TPFLG  001157          1244#   4752    4814
$TPS    001150          1239#   3168    4801    4814
$TRAP   032114          1721    5310#
$TRAP2  032136          5321#   5332
$TRP  = 000013          5325#   5334#   5335#   5336#   5337#   5338#   5339    5340#   5341    5342#   5343#   5344#   5345#
$TRPAD  032150          5315    5332#
$TSTM   001004          1204#
$TSTNM  001102          1217#   5042    5068
$TTYIN  030464          1789*   1791    2983    4932    4933    4955    4959#   5144
$TYPBN= ****** U        5338
$TYPDS  031442          5170#   5337
$TYPE   027412          4752#   4999    5325    5333
$TYPEC  027624          3283    4782    4789    4796    4801#   4802    4876
$TYPEX  027672          4807    4809    4812#
$TYPOC  031712          5255#   5334
$TYPON  031726          5254    5257#   5336
$TYPOS  031666          5250#   5335
$UNIT   001202          1260#
$UNITM  001010          1206#
$USWR   001214          1267#
$VECT1  001240          1292#
$VECT2  001242          1293#
$OFILL  032111          5251*   5255*   5265    5300#
$40CAT= ****** U        5052
```

```
.      = 032276          1162#   1166#   1170#   1173#   1176#   1178#   1180#   1185#   1191    1192#   1194#   1196#   1214#
                         1249    1613#   1614#   1631#   1632#   1642#   1716    2910    2957    3009#   3346    3372    3401
                         3415    3437    4577    4589    4605    4618    4721    4734#   4814    4817    4959#   4960    4967#
                         5020#   5068    5123#   5224#   5346#
.$ASTA= ****** U          4972    4975
.$X    = 001000          1191#   1196
```

```
COMMEN      1#    906#
ENDCOM      1#    906#
ERROR     800#   1890   1908   1913   1943   2001   2004   2013   2022   2030   2033   2049   2052   2055   2071
         2074   2077   2086   2097   2102   2113   2118   2132   2137   2168   2173   2176   2179   2209   2244
         2295   2301   2316   2357   2404   2442   2480   2507   2515   2520   2534   2547   2554   2559   2615
         2634   2720   2745   2785   2790   2794   2807   2826   2831   2843   2855   2860   2871   2897   2906
         2911   2936   2941   2946   3477   3525   3577   3614   3641   3667   3694   3740   3770   3855   3860
         3870   3872   3933   3942   3946   3956   3958   3965   4025   4032   4035   4040   4047   4245   4250
         4260   4262   4322   4331   4335   4345   4347   4414   4418   4423   4429   4546   4587   4603   4616
         4686   4705   4719
ESCAPE      1#    906#
GETPRI      1#    906#
GETSWR      1#    906#  1749#
MULT        1#    906#
NEWTST      1#    906#
POP         1#    906#  5014   5015   5211
PUSH        1#    906#  4975   4977   4998   5170
REPORT      1#    772#   906#
SCOPE     801#
SETPRI      1#    906#
SETTRA   5325#   5334   5335   5336   5337   5339   5341   5342   5343   5344
SETUP       1#    906#  1710
SKIP        1#    906#
SLASH       1#    906#
SPACE     906#
STARS       1#    906#   910    912    933    935    975    977   1028   1030   1061   1063   1123   1125   1158
         1160   1188   1190   1197   1210   1249   1252   1599   1601   1617   1619   1634   1640   1665   1667
         1777   1779   1785   1787   1797   1799   1808   1810   1823   1825   1861   1869   1880   1882   1895
         1897   1925   1927   1960   1962   1979   1981   1988   1990   2005   2007   2014   2016   2023   2025
         2035   2037   2057   2059   2078   2080   2089   2091   2103   2105   2121   2123   2138   2145   2195
         2197   2219   2221   2262   2264   2328   2330   2367   2369   2414   2416   2452   2454   2490   2497
         2571   2582   2675   2683   2763   2770   2810   2815   2876   2882   2913   2919   2962   2964   3012
         3014   3021   3023   3043   3045   3057   3059   3085   3087   3116   3118   3145   3147   3161   3165
         3174   3177   3209   3211   3257   3259   3305   3312   3327   3331   3354   3358   3383   3387   3409
         3411   3418   3422   3446   3448   3462   3464   3485   3488   3510   3512   3533   3536   3560   3563
         3570   3572   3603   3605   3630   3632   3656   3658   3683   3685   3709   3711   3756   3763   3795
         3798   3820   3822   3836   3842   3893   3895   3906   3912   3974   3977   3989   3994   4059   4064
         4099   4101   4130   4132   4141   4143   4153   4155   4161   4163   4170   4173   4185   4188   4214
         4216   4226   4232   4282   4284   4295   4301   4357   4360   4371   4376   4439   4444   4480   4482
         4508   4510   4530   4536   4559   4565   4649   4651   4673   4676   4691   4694   4710   4712   4730
         4732   4737   4816   4819   4895   4924   4970   5027   5070   5160   5227   5304
SWRSU       1#    906#  1723#
TRMTRP   5325#
TYPBIN      1#    906#
TYPDEC      1#    906#  2971   2975   2979   2986   2990   3038   4078   4458   5110
TYPNAM      1#    906#  1744
TYPNUM      1#    906#
TYPOCS      1#    906#
TYPOCT      1#    906#  3262   4837   5082   5107
TYPTXT      1#    906#
$$CMRE   1208#
$$CMTM   1208#
$$ESCA      1#    906#
$$NEWT      1#    906#
$$SET    5325#   5334   5335   5336   5337   5339   5341   5342   5343   5344
$$SETM   1739#
```

```
$$SKIP      1#     906#
.EQUAT      1#     772#     796
.HEADE      1#     772#     773
.KT11       1#
.SETUP      1#     772#    1678
.SWRHI      1#     772#
.$ACT1      1#     772#
.$APTB      1#     772#    1250#
.$APTH      1#     772#    1186
.$APTY      1#     772#    4968
.$ASTA      1#
.$CATC      1#     772#
.$CMTA      1#     772#    1208
.$DB2D      1#
.$DB2O      1#
.$DIV       1#
.$EOP       1#     772#
.$ERRO      1#     772#    5025
.$ERRT      1#     772#    5068
.$MULT      1#
.$POWE      1#
.$RAND      1#
.$RDDE      1#
.$RDOC      1#
.$READ      1#     772#    4814
.$R2AZ      1#
.$SAVE      1#
.$SB2D      1#
.$SB2O      1#
.$SCOP      1#     772#
.$SIZE      1#
.$SUPR      1#
.$TRAP      1#     772#    5302
.$TYPB      1#
.$TYPD      1#     772#    5158
.$TYPE      1#     772#    4735
.$TYPO      1#     772#    5225
.$40CA      1#
.1170       1#


. ABS.   032276     000


 ERRORS DETECTED:  0

 CVKDAD,CVKDAD.SEQ/CRF/SOL=SYSMAC.SML,CVKDAD.P11
 RUN-TIME: 55 62 5 SECONDS
 RUN-TIME RATIO: 229/123=1.8
 CORE USED:  33K  (65 PAGES)
```