

**IDENTIFICATION**

-----

PRODUCT CODE: MAINDEC-11-DVAAA-A-D  
PRODUCT NAME: AAV11 DIAGNOSTIC TEST  
DATE: OCTOBER 1976  
MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1976  
DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED UNDER A LICENSE FOR USE ONLY ON A SINGLE COMPUTER SYSTEM AND MAY BE COPIED ONLY WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE, OR ANY OTHER COPIES THEREOF, MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY OTHER PERSON EXCEPT FOR USE ON SUCH SYSTEM AND TO ONE WHO AGREES TO THESE LICENSE TERMS. TITLE TO AND OWNERSHIP OF THE SOFTWARE SHALL AT ALL TIMES REMAIN IN DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.



## TABLE OF CONTENTS

1.0	ABSTRACT
2.0	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
3.0	LOADING PROCEDURE
4.0	STARTING PROCEDURE
4.1	PROGRAM START
5.0	SOFTWARE SWITCH REGISTER
5.1	OPTIONS
5.2	CONTROL
6.0	ERROR REPORTING
6.1	ERROR COMMENT
6.2	ERROR DATA
7.0	MISCELLANEOUS
7.1	AAV11 BUS ADDRESS MODIFICATION
7.2	XXDP/APT NOTES
7.3	POWER FAIL
7.4	MULTIPLE AAV11 INTERFACE TESTING
7.5	RESTRICTIONS
8.0	EXECUTION TIME
9.0	PROGRAM TEST DESCRIPTIONS
9.1	LOGIC TEST
9.2	RAMP LOOP
9.3	STATIC CALIBRATION
9.4	DYNAMIC CALIBRATION
9.5	EXTENDED UNITS
9.6	TESTER MODE
10.0	LISTING

1.0 ABSTRACT  
-----

THE AAV11 DIAGNOSTIC PROGRAM IS A SERIES OF TESTS DESIGNED TO TEST ALL LOGIC FUNCTIONS AND DATA PATHS ACCESSIBLE. TOTAL PROGRAM CONTROL IS ACCOMPLISHED THRU THE CONSOLE TERMINAL VIA THE ODT/CONSOLE MICROCODE AND THE PROVISIONS OF SECTION 5.

2.0 REQUIREMENTS  
-----

2.1 EQUIPMENT

1. PDP11/03 COMPUTER OR LSI-11 PROCESSOR
2. DLV11 WITH I/O TYPE TERMINAL
3. AAV11 DAC OPTION

2.2 STORAGE

THE PROGRAM USES THE LOWER 4K OF MEMORY.

3.0 LOADING PROCEDURE  
-----

1. ASSURE THAT THE LSI-11 IS IN THE ODT MICROCODE STATE.
2. LOAD THE LOW OR HIGH SPEED READER WITH THE ABSOLUTE LOADER TAPE.
3. TYPE THE READER'S CSR ADDRESS (177560-LOW OR 177550-HIGH) AND CHARACTER 'L'.
4. AFTER TAPE IS LOADED, LOAD THE AAV11 BINARY TAPE INTO THE READER AND TYPE THE CHARACTER 'P'.
5. IF THE ABSOLUTE LOADER HAS ALREADY BEEN LOADED (STEPS 2 & 3), THEN ONLY THE STARTING ADDRESS OF THE ABSOLUTE LOADER AND THE CHARACTER 'G' NEED BE TYPED (WITH THE AAV11 BINARY TAPE IN THE APPROPRIATE READER).

4.0 STARTING PROCEDURE  
-----

1. MAKE SURE THE DEVICE BUS ADDRESS AGREE WITH THE DEFAULT VALUES DEFINED IN SECTION 7.1. IF NOT, CHANGE LOCATION(S) AS DESIRED VIA THE "ADDRESS/" ODT COMMAND.
2. INSURE THAT THE HALT SWITCH IS DISABLED (IF ANY).
4. TYPE THE STARTING ADDRESS OF 200 AND THE CHARACTER G.
5. THE PROGRAM WILL RESPOND BY TYPING THE PROGRAM TITLE.

4.1 PROGRAM START  
-----

200	STARTING ADDRESS OF THE LOGIC TEST <WITH UP TO FOUR AAV11'S>
204	STARTING ADDRESS OF THE RAMP LOOP
210	STARTING ADDRESS OF THE STATIC CALIBRATION
214	STARTING ADDRESS OF THE DYNAMIC CALIBRATION
230	STARTING ADDRESS OF THE LOGIC TEST <WITH UP TO SIXTEEN AAV11'S>
240	STARTING ADDRESS FOR THE OPTION TESTER.

5.0 SOFTWARE SWITCH REGISTER

-----

5.1 OPTIONS

SWITCH	OCTAL	FUNCTION
-----	-----	-----
SW15=1	100000	HALT ON ERROR
SW14=1	040000	LOOP ON TEST
SW13=1	020000	INHIBIT ERROR TYPEOUTS
SW11=1	004000	INHIBIT ITERATIONS
SW10=1	002000	BELL ON ERROR
SW09=1	001000	LOOP ON ERROR
SW08=1	0004XX	LOOP ON TEST IN SWR <7-0>

5.2 CONTROL

1. THE SOFTWARE SWITCH REGISTER 'SWREG' (LOC. 176) CAN BE CHANGED BY USING THE ODT FACILITIES.
2. THE SOFTWARE SWITCH REGISTER CAN BE CHANGED UNDER PROGRAM CONTROL BY TYPING THE 'CONTROL & G' KEYS. THIS KEYBOARD OPERATION WILL PRINT OUT THE CURRENT CONTENTS AND ACCEPT NEW OCTAL SWITCH REGISTER DATA TERMINATED WITH A CARRIAGE RETURN.
3. ONCE THE ODT MODE HAS BEEN ENTERED BECAUSE OF AN ERROR CONDITION WITH BIT15 SET (HALT ON ERROR), STEP #2 ABOVE IS OF NO VALUE, SO RESORT TO STEP #1 TO ALTER THE SOFTWARE SWITCH REGISTER IF DESIRED BEFORE TYPING 'P' (CONTINUE).
4. IF THE PROGRAM IS PERFORMING RESET INSTRUCTIONS, SEVERAL 'CONTROL & G' COMMANDS MAY BE NECESSARY TO BE ACKNOWLEDGE BY THE PROGRAM.

6.0 ERROR REPORTING

-----

6.1 ERROR COMMENT

ALL ERRORS ARE ACCOMPANIED WITH AN ENGLISH LANGUAGE DESCRIPTIVE COMMENT AS TO THE TYPE OF FAILURE. FURTHER QUALIFICATION OF THE ERROR CAN BE OBTAINED IF NEEDED FROM THE COMMENT AT THE ERROR PC OR FROM THE TEST ITSELF.

6.2 ERROR DATA

*ERRPC	LISTING ADDRESS WHERE THE ERROR WAS DETECTED
*BUSADR	AAV11 BUS REG ADDRESS OF CONCERNED OPERATION
EXPCT	DATA THAT WAS EXPECTED
RCVD	DATA THAT WAS RECEIVED

\*ALWAYS REPORTED

7.0 MISCELLANEOUS

-----

7.1 AAV11 BUS ADDRESS MODIFICATION

MODIFY LOCATION 'SBASE' (LOC. 1250) IF BASE BUS ADDRESS IS NOT 170440.

\*NOTE: USE THE LSI-11 ODT FACILITIES TO MODIFY THIS LOCATIONS AFTER PROGRAM LOAD.

7.2 XXDP/APT NOTES

THIS DIAGNOSTIC IS CHAINABLE UNDER XXDP (REQUIRES 8K OR MORE). THIS DIAGNOSTIC DOES SUPPORT "APT" BUT HAS NOT BEEN RUN UNDER IT.

7.3 POWER FAIL

A POWER FAILURE WILL CAUSE A RESTART MESSAGE ON POWER UP AT WHICH TIME THE PROGRAM IS RESTARTED (ONLY ON SYSTEMS WITH NON-VOLATILE MEMORY AND WITH APPROPRIATE HARDWARE).

7.4 MULTIPLE AAV11 INTERFACE TESTING

THIS PROGRAM DOES "AUTO-SIZE" THE NUMBER OF AAV11'S CONNECTED. THIS DIAGNOSTIC WILL TEST SEQUENTIALLY UP TO 4 AAV11 INTERFACES, WHEN STARTED AT 200 AND 16. WHEN STARTED AT ADDRESS 230, WITH CONTIGUOUS BUS ADDRESSES, THE "AUTO-SIZE" CAN BE INHIBITED BY THE OPERATOR SETTING BIT 15 OF LOCATION "SENV (LOC. 1214) AND LOADING LOCATION 'SBASE' WITH THE ADDRESS OF THE ONE UNIT TO BE TESTED.

7.5 RESTRICTIONS

NONE

8.0 EXECUTION TIME

-----

EXECUTION TIME RANGES FROM ABOUT 5 SECONDS WITH NO ITERATIONS TO ABOUT 20 SECONDS WITH ITERATIONS ENABLED WITH ONE AAV11 CONNECTED. AN END PASS MESSAGE INDICATES ALL TESTS HAVE COMPLETED ON ALL SELECTED UNITS. END OF PASS WILL ALSO REPORT TOTAL ERROR COUNT AND ANY UNIT'S THAT HAD ERRORED.

9.0 PROGRAM TEST DESCRIPTIONS

-----

9.1 LOGIC TESTS (SA 200)

THIS DIAGNOSTIC CONTAINS A SERIES OF INDEPENDENT TESTS DESIGNED TO TEST LOGIC FUNCTIONS AND DATA PATHS OF THE AAV11 DAC CONTROL. A COMPLETE LIST OF TESTS IS AVAILABLE IN THE TABLE OF CONTENTS AT THE BEGINNING OF THE LISTING. THE COMMENT FIELD WITHIN EACH TEST CAN BE BENEFICIAL IN TEST UNDERSTANDING. WHEN STARTED AT LOCATION 200, THE PROGRAM WILL AUTO-SIZE UP TO 4 AAV11'S TO BE TESTED.

9.2 RAMP LOOP (SA 204)

THIS LOOP IS PROVIDED A METHOD FOR THE OPERATOR TO INSPECT AND VERIFY ANALOG OPERATION OF ALL DAC BITS. THE LOOP ALSO ENABLES THE OPERATOR TO VERIFY THAT NO TWO DAC'S ARE INTERCONNECTED.

9.3 STATIC CALIBRATION LOOP (SA 210)

THIS LOOP PROVIDES THE OPERATOR WITH A SIMPLE LOOP FOR VERIFYING THE INDIVIDUAL DAC BITS AND THE OPERATION OF DAC #3 DIGITAL OUTPUT BITS. THE VALUE OF THE SWITCH REGISTER IS LOADED INTO ALL DAC'S AND THE OUTPUT VOLTAGE CAN BE MONITORED.

9.4 DYNAMIC CALIBRATION LOOP (SA 214)

THIS PROVIDES THE OPERATOR WITH A LOOP THAT LOADS THE VALUE OF THE SWITCH REGISTER INTO THE DAC'S AND THEN AFTER A DELAY CLEARS THE DAC REGISTERS. THIS PROVIDES A SWITCHING PATTERN BETWEEN THE SELECTED VOLTAGE AND 0.

9.5 EXTENDED UNITS (SA 230)

SAME FUNCTION AS LOGIC TEST BUT ON 16. AAV11'S

9.6 TESTER SUPPORT (SA 240)

INITIALL PERFORMS THE LOGIC TESTS AND THEN EMPLOYS A KNOWN GOOD A TO D CONVERTER TO AID IN ADJUSTING THE POT'S ON THE AAV11 BOARD. THE OPERATOR IS INFORMED AS TO WHICH POT TO ADJUST AND WHICH D TO A CONVERTER IS TESTED.

10.0 LISTING

-----

5551 BASIC DEFINITIONS  
5556 OPERATIONAL SWITCH SETTINGS  
5557 TRAP CATCHER  
(1) STARTING ADDRESS(ES)  
5571 ACT11 HOOKS  
5573 APT PARAMETER BLOCK  
5574 COMMON TAGS  
(2) APT MAILBOX=ETABLE  
(1) ERROR POINTER TABLE  
5696 INITIALIZE THE COMMON TAGS

5723 TEST # DESCRIPTION  
5724 -----  
5725 -----  
5726 -----

5741 DETERMINE THE NUMBER OF AAV11 ON THIS SYSTEM  
5783 T1 TEST THAT THE AAV11 RESPONDS TO THE CPU  
5795 T2 TEST THAT DAC0 REGISTER CAN BE CLEARED  
5803 T3 TEST THAT DAC0 REGISTER CAN BE LOADED WITH #7777  
5811 T4 TEST THAT DAC0 REGISTER CAN HOLD A FLOATING 1 PATTERN  
5821 T5 TEST THAT DAC0 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)  
5822 T6 TEST THE "SUB" INSTRUCTION WORKS ON DAC0  
5823 T7 TEST THAT DAC1 REGISTER CAN BE CLEARED  
5830 T10 TEST THAT DAC #1 REGISTER CAN BE LOADED WITH #7777  
5838 T11 TEST THAT DAC #1 REGISTER CAN HOLD A FLOATING 1 PATTERN  
5847 T12 TEST THAT DAC1 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)  
5848 T13 TEST THE "SUB" INSTRUCTION WORKS ON DAC1  
5850 T14 TEST THAT THE DAC #2 REGISTER CAN BE CLEARED  
5850 T15 TEST THAT THE DAC #2 REGISTER CAN BE LOADED WITH #7777  
5866 T16 TEST THAT THE DAC #2 REGISTER CAN HOLD A FLOATING 1 PATTERN  
5876 T17 TEST THAT DAC2 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)  
5878 T20 TEST THE "SUB" INSTRUCTION WORKS ON DAC2  
5880 T21 TEST THAT THE DAC #3 REGISTER CAN BE CLEARED  
5888 T22 TEST THAT THE DAC #3 REGISTER CAN BE LOADED WITH #7777  
5896 T23 TEST THAT THE DAC #3 REGISTER CAN HOLD A FLOATING 1 PATTERN  
5907 T24 TEST THAT DAC3 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)  
5908 T25 TEST THE "SUB" INSTRUCTION WORKS ON DAC3  
5910 T26 TEST THAT THE FOUR DAC REGISTERS CAN HOLD DIFFERENT DATA  
5939 T27 TEST THAT RESET CLEARS DAC #0 REGISTER  
5948 T30 TEST THAT RESET CLEARS DAC #1 REGISTER  
5958 T31 TEST THAT RESET CLEARS DAC #2 REGISTER  
5966 T32 TEST THAT RESET CLEARS DAC #3 REGISTER  
5977 T33 DETERMINE IF MORE AAV11'S REMAIN TO BE TESTED  
6015 T34 DETERMINE IF RUNNING ON THE HARDWARE TESTER (IF NOT REPORT END OF PASS)  
6020 T35 TEST THAT DAC #3 OUTPUT BITS (0-3) FUNCTION  
6035 T36 VERIFY THE AAV11 +15 SUPPLY  
6044 T37 VERIFY THE AAV11 -15 SUPPLY  
6054 T40 DAC0 OFFSET ADJUSTMENT  
(4) T41 DAC0 GAIN ADJUSTMENT  
(4) T42 DAC0 CALIBRATION  
6056 T43 DAC1 OFFSET ADJUSTMENT  
(4) T44 DAC1 GAIN ADJUSTMENT  
(4) T45 DAC1 CALIBRATION  
6058 T46 DAC2 OFFSET ADJUSTMENT  
(4) T47 DAC2 GAIN ADJUSTMENT  
(4) T50 DAC2 CALIBRATION  
6060 T51 DAC3 OFFSET ADJUSTMENT

(4)	T52	DAC3 GAIN ADJUSTMENT
(4)	T53	DAC3 CALIBRATION
6062		END OF PASS ROUTINE
6073		SUBROUTINE TO ADJUST THE DAC'S OFFSET POTS
6102		SUBROUTINE TO ADJUST THE GAIN ADJUSTMENT POTS
6128		SUBROUTINE TO TEST THE D/A CALIBRATION
6152		SUBROUTINE TO LOAD A VOLTAGE INTO THE VOLTAGE SOURCE
6175		SUBROUTINE TO CONVERT CHANNEL N ON THE TESTER A/D
6200		SUBROUTINE TO LOOP UNTIL OPERATOR TYPES AN "SPACE"
6222		SUBROUTINE TO COMPARE TWO LOCATIONS BY THE SPREAD
6243		
6244		DAC ADJUSTMENT ROUTINES
6245		-----
6246		
6248		FULL SCALE RAMP ON EACH RAMP
6269		STATIC DAC CALIBRATION
6281		DYNAMIC DAC CALIBRATION
6318		
6319		MISC. SUB-ROUTINES, ASCII MESSAGES AND SOFTWARE HANDLERS
6320		
6325		ASCII MESSAGES
6389		BINARY TO ASCII AND TYPE ROUTINE
6390		CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
6391		SCOPE HANDLER ROUTINE
6392		ERROR HANDLER ROUTINE
6393		ERROR MESSAGE TIMEOUT ROUTINE
6394		POWER DOWN AND UP ROUTINES
6398		BINARY TO OCTAL (ASCII) AND TYPE
6399		TYPE ROUTINE
6400		TTY INPUT ROUTINE
6401		READ AN OCTAL NUMBER FROM THE TTY
6402		APT COMMUNICATIONS ROUTINE
6405		TRAP DECODER
(3)		TRAP TABLE



```

5550 .TITLE MAINDEC-11-DVAAA-A      AAV11  DIAGNOSTIC
(1)  ;*COPYRIGHT (C) 1976
(1)  ;*DIGITAL EQUIPMENT CORP.
(1)  ;*MAYNARD, MASS. 01754
(1)  ;*
(1)  ;*PROGRAM BY RAYMOND SHODD
(1)  ;*
(1)  ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSHAC
(1)  ;*PACKAGE (MAINDEC-11-D2BAC-C1), MAR 26, 1976.
(1)  ;*
5551 .SBTTL BASIC DEFINITIONS
(1)  ;*INITIAL ADDRESS OF THE STACK POINTER *** 11PR ***
(1)  STACK= 11PR
(1)  .EQUIV EMT,ERROR      ;*BASIC DEFINITION OF ERROR CALL
(1)  .EQUIV IOT,SCOPE     ;*BASIC DEFINITION OF SCOPE CALL
(1)  ;*
(1)  ;*MISCELLANEOUS DEFINITIONS
(1)  MT= 11              ;*CODE FOR HORIZONTAL TAB
(1)  LF= 12              ;*CODE FOR LINE FEED
(1)  CR= 15              ;*CODE FOR CARRIAGE RETURN
(1)  CRLF= 200           ;*CODE FOR CARRIAGE RETURN-LINE FEED
(1)  PS= 177776         ;*PROCESSOR STATUS WORD
(1)  .EQUIV PS,PSW
(1)  177774              ;*STACK LIMIT REGISTER
(1)  177772              ;*PROGRAM INTERRUPT REQUEST REGISTER
(1)  177570              ;*HARDWARE SWITCH REGISTER
(1)  177570              ;*HARDWARE DISPLAY REGISTER
(1)  ;*
(1)  ;*GENERAL PURPOSE REGISTER DEFINITIONS
(1)  R0= X0              ;*GENERAL REGISTER
(1)  R1= X1              ;*GENERAL REGISTER
(1)  R2= X2              ;*GENERAL REGISTER
(1)  R3= X3              ;*GENERAL REGISTER
(1)  R4= X4              ;*GENERAL REGISTER
(1)  R5= X5              ;*GENERAL REGISTER
(1)  R6= X6              ;*GENERAL REGISTER
(1)  R7= X7              ;*GENERAL REGISTER
(1)  .EQUIV R6,SP       ;*STACK POINTER
(1)  .EQUIV R7,PC       ;*PROGRAM COUNTER
(1)  ;*
(1)  ;*PRIORITY LEVEL DEFINITIONS
(1)  PR0= 0              ;*PRIORITY LEVEL 0
(1)  PR1= 40             ;*PRIORITY LEVEL 1
(1)  PR2= 100           ;*PRIORITY LEVEL 2
(1)  PR3= 140           ;*PRIORITY LEVEL 3
(1)  PR4= 200           ;*PRIORITY LEVEL 4
(1)  PR5= 240           ;*PRIORITY LEVEL 5
(1)  PR6= 300           ;*PRIORITY LEVEL 6
(1)  PR7= 340           ;*PRIORITY LEVEL 7
(1)  ;*
(1)  ;*SWITCH REGISTER* SWITCH DEFINITIONS
(1)  SW15= 100000
(1)  SW16= 40000
(1)  SW13= 20000
(1)  SW12= 10000

```

```

(1)  SW11= 4000
(1)  SW10= 2000
(1)  SW09= 1000
(1)  SW08= 400
(1)  SW07= 200
(1)  SW06= 100
(1)  SW05= 40
(1)  SW04= 20
(1)  SW03= 10
(1)  SW02= 4
(1)  SW01= 2
(1)  SW00= 1
(1)  .EQUIV SW00,SW0
(1)  .EQUIV SW08,SW8
(1)  .EQUIV SW07,SW7
(1)  .EQUIV SW06,SW6
(1)  .EQUIV SW05,SW5
(1)  .EQUIV SW04,SW4
(1)  .EQUIV SW03,SW3
(1)  .EQUIV SW02,SW2
(1)  .EQUIV SW01,SW1
(1)  .EQUIV SW00,SW0
(1)  ;*
(1)  ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
(1)  BIT15= 100000
(1)  BIT14= 40000
(1)  BIT13= 20000
(1)  BIT12= 10000
(1)  BIT11= 4000
(1)  BIT10= 2000
(1)  BIT09= 1000
(1)  BIT08= 400
(1)  BIT07= 200
(1)  BIT06= 100
(1)  BIT05= 40
(1)  BIT04= 20
(1)  BIT03= 10
(1)  BIT02= 4
(1)  BIT01= 2
(1)  BIT00= 1
(1)  .EQUIV BIT00,BIT0
(1)  .EQUIV BIT06,BIT6
(1)  .EQUIV BIT07,BIT7
(1)  .EQUIV BIT08,BIT8
(1)  .EQUIV BIT09,BIT9
(1)  .EQUIV BIT10,BIT10
(1)  .EQUIV BIT11,BIT11
(1)  .EQUIV BIT12,BIT12
(1)  .EQUIV BIT13,BIT13
(1)  .EQUIV BIT14,BIT14
(1)  .EQUIV BIT15,BIT15
(1)  ;*
(1)  ;*BASIC "CPU" TRAP VECTOR ADDRESSES
(1)  TRTVEC= 14         ;*TRACE TRAP
(1)  TRIVEC= 14
(1)  RESVEC= 10        ;*RESERVED AND ILLEGAL INSTRUCTIONS
(1)  000014            ;*?" BIT
(1)  000014            ;*
(1)  000014            ;*

```

```

(1) 000014 BPTVEC= 14 //BREAKPOINT TRAP (BPT)
(1) 000020 IOTVEC= 20 //INPUT/OUTPUT TRAP (IOT) **SCOPE**
(1) 000024 PWRVEC= 24 //POWER FAIL
(1) 000030 EMTVEC= 30 //EMULATOR TRAP (EMT) **ERROR**
(1) 000034 TRAPVEC=34 //TRAP TRAP
(1) 000060 TKVEC= 60 //TTY KEYBOARD VECTOR
(1) 000064 TPVEC= 64 //TTY PRINTER VECTOR
(1) 000240 PIRGVEC=240 //PROGRAM INTERRUPT REQUEST VECTOR
5552
5553 170440 ABASE=170440
    
```

```

5555
5556 .SBTTL OPERATIONAL SWITCH SETTINGS
(1) ?*
(1) ?* SWITCH USE
(1) ?* -----
(1) ?* 15 HALT ON ERROR
(1) ?* 14 LOOP ON TEST
(1) ?* 13 INHIBIT ERROR TYPEOUTS
(1) ?* 11 INHIBIT ITERATIONS
(1) ?* 10 BELL ON ERROR
(1) ?* 9 LOOP ON ERROR
(1) ?* 8 LOOP ON TEST IN SWR<710>
5557 .SBTTL TRAP CATCHER
(1) .#P
(1) 000000 //ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ",*2,HALT"
(1) //SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
(1) //LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
(1) .#174
(1) 000174 000000 DISPREG: ,WORD 0 //SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SWREG: ,WORD 0 //SOFTWARE SWITCH REGISTER
(1) .SBTTL STARTING ADDRESS(ES)
(1) JMP #RREGIN //JUMP TO STARTING ADDRESS OF PROGRAM
5558 000200 000137 001450 JMP FULLRMP //JUMP TO FULL RAMP LOOP
5559 000210 000137 006570 JMP STATIC //JUMP TO STATIC DAC CALIBRATION
5560 000214 000137 006716 JMP DYNCAL //JUMP TO DYNAMIC DAC CALIBRATION
5561
5562 .#230
5563 000230 000137 001440 JMP A000K //JUMP AND ENABLE EXTENDED UNITS <16,>
5564
5565 .#240
5566 000240 000137 001432 JMP TESTER //JUMP TO TESTER SA.
5567
5568 .#100
5569 000100 000104 000200 000002 JMP 100,200,2 //EVENT SAFE GUARD
    
```

```

5571 .SBTTL ACT11 HOOKS
(1)
(2)
(1) *****
(1) HOOKS REQUIRED BY ACT11
(1) $SVPCL. $SAVE PC
(1) $=6 SENDAD $SET LOC,46 TO ADDRESS OF SENDAD IN $SEOP
(1) 000246 005542 $=52
(1) 000752 000000 $WORD 0 $SET LOC,52 TO ZERO
(1) 000106 001000 $=1000 $RESTORE PC
5572
5573 .SBTTL APT PARAMETER BLOCK
(1)
(2) *****
(1) $SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(1) *****
(1) $X=. $SAVE CURRENT LOCATION
(1) 001000 $=24 $SET POWER FAIL TO POINT TO START OF PROGRAM
(1) 00024 000200 200 $FOR APT START UP
(1) 00044 001000 $=44 $POINT TO APT INDIRECT ADDRESS PNTR.
(1) 000244 001000 $APTHDR $POINT TO APT HEADER BLOCK
(1) $=5X $RESET LOCATION COUNTER
(2) *****
(1) $SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1) $INTERFACE SPEC.
(1)
(1)
(1) $APTHDR: $WORD 0
(1) 001000 000000 $SHRSTS: $WORD 0 $TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 001002 001174 $MADR: $WORD $MAIL $ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 001004 000030 $STMT: $WORD 30 $RUN TIM OF LONGEST TEST
(1) 001006 000010 $PASTM: $WORD 10 $RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 001010 000030 $UNITM: $WORD 30 $ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 001012 000031 $WORD $ETEND-$MAIL/2 $LENGTH MAILBOX-$TABLE($WORDS)

```

```

5574 .SBTTL COMMON TAGS
(1)
(2) *****
(1) $THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1) $USED IN THE PROGRAM.
(1)
(1) $=1100
(1) 001100 001100 $CMTAG: $WORD 0 $START OF COMMON TAGS
(1) 001100 000000 $STNM: $BYTE 0 $CONTAINS THE TEST NUMBER
(1) 001102 000 $ERFLG: $BYTE 0 $CONTAINS ERROR FLAG
(1) 001103 000 $ICNT: $WORD 0 $CONTAINS SUBTEST ITERATION COUNT
(1) 001104 000000 $LPADR: $WORD 0 $CONTAINS SCOPE LOOP ADDRESS
(1) 001106 000000 $LPERR: $WORD 0 $CONTAINS SCOPE RETURN FOR ERRORS
(1) 001110 000000 $ERTTL: $WORD 0 $CONTAINS TOTAL ERRORS DETECTED
(1) 001112 000000 $ITEMR: $BYTE 0 $CONTAINS ITEM CONTROL BYTE
(1) 001114 000 $SERMAX: $BYTE 1 $CONTAINS MAX. ERRORS PER TEST
(1) 001115 001 $ERRPC: $WORD 0 $CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001116 000000 $GDADR: $WORD 0 $CONTAINS ADDRESS OF 'GOOD' DATA
(1) 001120 000000 $BDADR: $WORD 0 $CONTAINS ADDRESS OF 'BAD' DATA
(1) 001124 000000 $GDDAT: $WORD 0 $CONTAINS 'GOOD' DATA
(1) 001126 000000 $RDDAT: $WORD 0 $CONTAINS 'BAD' DATA
(1) 001130 000000 $WORD 0 $RESERVED--NOT TO BE USED
(1) 001132 000000 $WORD 0
(1) 001134 000 $AUTOB: $BYTE 0 $AUTOMATIC MODE INDICATOR
(1) 001135 000 $INTAG: $BYTE 0 $INTERRUPT MODE INDICATOR
(1) 001136 000000 $WORD 0
(1) 001140 177570 $SWR: $WORD $DSWR $ADDRESS OF SWITCH REGISTER
(1) 001142 177570 $DISPLA: $WORD $DISP $ADDRESS OF DISPLAY REGISTER
(1) 001144 177560 $TKS: 177560 $TTY KBD STATUS
(1) 001146 177562 $TKR: 177562 $TTY KBD BUFFER
(1) 001150 177564 $TPSI: 177564 $TTY PRINTER STATUS REG. ADDRESS
(1) 001152 177566 $TPRI: 177566 $TTY PRINTER BUFFER REG. ADDRESS
(1) 001154 000 $NULL: $BYTE 0 $CONTAINS NULL CHARACTER FOR FILL
(1) 001155 002 $FILLS: $BYTE 2 $CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001156 012 $FILLC: $BYTE 12 $INSERT FILL CHARS. AFTER A 'LINE FEED'
(1) 001157 000 $TPFLG: $BYTE 0 $'TERMINAL AVAILABLE' FLAG (BIT407=0=YES)
(1) 001160 000000 $TIMES: 0 $MAX. NUMBER OF ITERATIONS
(1) 001162 000000 $ESCAPE: 0 $ESCAPE ON ERROR ADDRESS
(1) 001164 177607 000377 $BELL: $ASCIIZ <207><377><377> $CODE FOR BELL
(1) 001170 077 $QUES: $ASCII ?? $QUESTION MARK
(1) 001171 015 $CRLF: $ASCII <15> $CARRIAGE RETURN
(1) 001172 000012 $LF: $ASCIIZ <12> $LINE FEED
(2) *****
(2) .SBTTL APT MAILBOX-$TABLE
(2)
(2) *****
(2) $EVEN
(2) 001174 000000 $MAIL: $WORD $APT MAILBOX
(2) 001174 000000 $MSGTY: $WORD $MSGTY $MESSAGE TYPE CODE
(2) 001176 000000 $SFATAL: $WORD $AFATAL $FATAL ERROR NUMBER
(2) 001200 000000 $STESTN: $WORD $ATESTN $TEST NUMBER
(2) 001202 000000 $SPASS: $WORD $APASS $PASS COUNT
(2) 001204 000000 $DEVCT: $WORD $ADEVCT $DEVICE COUNT
(2) 001206 000000 $SUNIT: $WORD $AUNIT $I/O UNIT NUMBER
(2) 001210 000000 $MSGAD: $WORD $MSGAD $MESSAGE ADDRESS

```

```
(2) 001212 000000 SMSGLG1 ,WORD AMAGLG 11MESSAGE LENGTH
(2) 001214 000000 SETABLE1 11APT ENVIRONMENT TABLE
(2) 001214 000000 SENVM1 ,BYTE AENV 11ENVIRONMENT BYTE
(2) 001215 000000 SENVM1 ,BYTE AENVH 11ENVIRONMENT MODE BITS
(2) 001216 000000 SSWREG1 ,WORD ASWREG 11APT SWITCH REGISTER
(2) 001220 000000 SUSWR1 ,WORD AUSWR 11USER SWITCHES
(2) 001222 000000 SCPUOP1 ,WORD ACPUPD 11CPU TYPE,OPTIONS
(2) 1115=11CPU TYPE
(2) 1111/04=01,11/05=02,11/20=03,11/40=04,11/05=05
(2) 1111/70=06,11/07=07,0=10
(2) 11BIT 10=REAL TIME CLOCK
(2) 11BIT 9=FLOATING POINT PROCESSOR
(2) 11BIT 8=MEMORY MANAGEMENT
(2) 001224 000000 SHAMS11 ,BYTE AMAMS1 11HIGH ADDRESS,M.3. BYTE
(2) 001225 000000 SHMTP11 ,BYTE AMTYP1 11MEM. TYPE,BLK#1
(2) 11MEM,TYPE BYTE -- (HIGH BYTE)
(2) 11400 NSEC CORE=001
(2) 11300 NSEC RIPOLAR=002
(2) 11500 NSEC MOS=003
(2) 001226 000000 SHADR11 ,WORD AMADR1 11HIGH ADDRESS,BLK#1
(2) 11MEM, LAST ADDR, #3 BYTES, THIS WORD AND LOW OF "TYPE" ABOVE
(2) 001230 000000 SHAMS21 ,BYTE AMAMS2 11HIGH ADDRESS,M.3. BYTE
(2) 001231 000000 SHMTP21 ,BYTE AMTYP2 11MEM,TYPE,BLK#2
(2) 001232 000000 SHADR21 ,WORD AMADR2 11MEM, LAST ADDRESS,BLK#2
(2) 001234 000000 SHAMS31 ,BYTE AMAMS3 11HIGH ADDRESS,M.3. BYTE
(2) 001235 000000 SHMTP31 ,BYTE AMTYP3 11MEM,TYPE,BLK#3
(2) 001236 000000 SHADR31 ,WORD AMADR3 11MEM, LAST ADDRESS,BLK#3
(2) 001240 000000 SHAMS41 ,BYTE AMAMS4 11HIGH ADDRESS,M.3. BYTE
(2) 001241 000000 SHMTP41 ,BYTE AMTYP4 11MEM,TYPE,BLK#4
(2) 001242 000000 SHADR41 ,WORD AMADR4 11MEM, LAST ADDRESS,BLK#4
(2) 001244 000000 SVECT11 ,WORD AVECT1 11INTERRUPT VECTOR#1,BUS PRIORITY#1
(2) 001246 000000 SVECT21 ,WORD AVECT2 11INTERRUPT VECTOR#2BUS PRIORITY#2
(2) 001250 170440 SHASE1 ,WORD ABASE 11BASE ADDRESS OF EQUIPMENT UNDER TEST
(2) 001252 000000 SDEVH1 ,WORD ADEVH 11DEVICE MAP
(2) 001254 000000 SCOW11 ,WORD ACD#1 11CONTROLLER DESCRIPTION WORD#1
(2) 001256 000000 SETEND1 ,MEXIT
```

```
(1) .S07TL ERROR POINTER TABLE
(1)
(1) 11THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1) 11THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1) 11LOCATION SITE#B. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1) 11NOTE1: IF SITE#B IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
(1) 11NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1)
(1) 11* EM //POINTS TO THE ERROR MESSAGE
(1) 11* DM //POINTS TO THE DATA HEADER
(1) 11* DT //POINTS TO THE DATA
(1) 11* DF //POINTS TO THE DATA FORMAT
(1)
(1) 001256 5575 5576 5577 5578 5579 5580 5581 5582 5583 5584 5585 5586 5587 5588 5589 5590 5591 5592 5593 5594 5595 5596 5597 5598 5599 5600 5601 5602 5603 5604 5605 5606 5607 5608 5609 5610 5611 5612 5613
(1) 001256 007122 010322 011112 011204 007176 010267 011120 011204 007225 010267 011132 011204 007254 010267 011144 011204 007303 010267 011156 011204 007332 010337 011170 011204
(1) SERRPC:
(1) ITEM 1
(1) EM1 //BUS TIME-OUT WHEN REF. A DAC ADDRESS
(1) DM2 //ERRPC BUSADR
(1) DT1 //ERRPC SBDDAT
(1) DF0
(1) ITEM 2
(1) EM2 //DAC #0 REGISTER IN ERROR
(1) DM1 //ERRPC BUSADR GOOD BAD
(1) DT2 //ERRPC DACR SGDDAT SBDDAT
(1) DF0
(1) ITEM 3
(1) EM3 //DAC #1 REGISTER IN ERROR
(1) DM1 //ERRPC BUSADR GOOD BAD
(1) DT3 //ERRPC DAC1 SGDDAT SBDDAT
(1) DF0
(1) ITEM 4
(1) EM4 //DAC #2 REGISTER IN ERROR
(1) DM1 //ERRPC BUSADR GOOD BAD
(1) DT4 //ERRPC DAC2 SGDDAT SBDDAT
(1) DF0
(1) ITEM 5
(1) EM5 //DAC #3 REGISTER IN ERROR
(1) DM1 //ERRPC BUSADR GOOD BAD
(1) DT5 //ERRPC DAC3 SGDDAT SBDDAT
(1) DF0
(1) ITEM 6
(1) EM6 //SELECTED DAC OFFSET POT IS NOT ADJUSTED CORRECTLY
(1) DM6 //ERRPC BUSADR EXPECT WAS SPREAD
(1) DT6 //ERRPC DACBAD SGDDAT SBDDAT SPREAD
(1) DF0
```

DVA AAA.P11		ERROR POINTER TABLE		ITEM	
5615				7	
5616	001336	007413		EM7	ISELECTED DAC GAIN POT IS NOT ADJUSTED CORRECTLY
5617	001340	010337		DM6	JERRPC BUSADR EXPECT WAS SPREAD
5618	001342	011170		DT6	JERRPC DACBAD SGDDAT SBDDAT SPREAD
5619	001344	011204		DF0	
5620					
5621				10	
5622	001346	007472		EM10	ISELECTED DAC HAS A LINEARITY PROBLEM
5623	001350	010337		DM6	JERRPC BUSADR EXPECT WAS SPREAD
5624	001352	011170		DT6	JERRPC DACBAD SGDDAT SBDDAT SPREAD
5625	001354	011204		DF0	
5626					
5627				11	
5628	001356	007537		EM11	+15 VOLT SUPPLY IS INCORRECT
5629	001360	010337		DM6	JERRPC BUSADR EXPECT WAS SPREAD
5630	001362	011170		DT6	JERRPC DACBAD SGDDAT SBDDAT SPREAD
5631	001364	011204		DF0	
5632					
5633				12	
5634	001366	007574		EM12	-15 VOLT SUPPLY IS INCORRECT
5635	001370	010337		DM6	JERRPC BUSADR EXPECT WAS SPREAD
5636	001372	011170		DT6	JERRPC DACBAD SGDDAT SBDDAT SPREAD
5637	001374	011204		DF0	
5638					
5639				13	
5640	001376	007631		EM13	DNAC #3 DIGITAL OUTPUT BITS IN ERROR
5641	001400	010267		DM1	JERRPC BUSADR GOOD BAD
5642	001402	011196		DT5	JERRPC DAC3 SGDDAT SBDDAT
5643	001404	011204		DF0	
5644					
5645				14	
5646	001406	007675		EM14	WAKE UP OPERATOR AND ADJUST THE POT
5647	001410	000000		0	
5648	001412	000000		0	
5649	001414	000000		0	
5650					
5651	001416	000010		VADDR1	10
5652	001420	000000		EVER:	0
5653	001422	170440		DAC0:	ABASE
5654	001424	170442		DAC1:	ABASE+2
5655	001426	170444		DAC2:	ABASE+4
5656	001430	170446		DAC3:	ABASE+6

DVA AAA.P11		ERROR POINTER TABLE		TESTER:	INC	WFTST	JINDICATE TESTER MODE
5687	001436	000411		AR		REGIN1	
5688	001440	012737	000021	ADDOK:	MOV	#17.,NUMBOK	JLOAD 16 MAX UNITS
5689	001446	000403		AR		REGINA	
5690	001450	012737	000005	BEGIN:	MOV	#5.,NUMBOK	JLOAD 4 MAX UNITS
5691	001456	005037	007020	BEGIN:	CLR	WFTST	
5692	001462	005037	007012	REGIN1:	CLR	TEMP	
5693	001466	005037	001420		CLR	EVER	
5694	001472	000005			RESET		
5696				.SRTTL		INITIALIZE THE COMMON TAGS	
(1)				;;CLEAR		THE COMMON TAGS (SCMTAG) AREA	
(1)	001474	012706	001100	MOV	#0CMTAG,R6	;;FIRST LOCATION TO BE CLEARED	
(1)	001500	005026		CLR	(R6)+	;;CLEAR MEMORY LOCATION	
(1)	001502	022706	001140	CMP	#SWR,R6	;;DONE?	
(1)	001506	001374		BNE	=6	;;LOOP BACK IF NO	
(1)	001510	012706	001100	MOV	#STACK,SP	;;SETUP THE STACK POINTER	
(1)				;;INITIALIZE A FEW VECTORS			
(1)	001514	012737	011524	MOV	#SCOPE,#IOTVEC	;;IOT VECTOR FOR SCOPE ROUTINE	
(1)	001522	012737	000340	MOV	#340,#IOTVEC+2	;;LEVEL 7	
(1)	001530	012737	012006	MOV	#ERROR,#EMTVEC	;;EMT VECTOR FOR ERROR ROUTINE	
(1)	001536	012737	000340	MOV	#340,#EMTVEC+2	;;LEVEL 7	
(1)	001544	012737	014412	MOV	#TRAP,#TRAPVEC	;;TRAP VECTOR FOR TRAP CALLS	
(1)	001552	012737	000340	MOV	#340,#TRAPVEC+2	;;LEVEL 7	
(1)	001560	012737	012336	MOV	#SPWRN,#PWRVEC	;;POWER FAILURE VECTOR	
(1)	001566	012737	000340	MOV	#340,#PWRVEC+2	;;LEVEL 7	
(1)	001574	005037	001160	CLR	STIMES	;;INITIALIZE NUMBER OF ITERATIONS	
(1)	001600	005037	001162	CLR	SESCAPE	;;CLEAR THE ESCAPE ON ERROR ADDRESS	
(1)	001604	112737	000001	MOV#	#1,SEMAX	;;ALLOW ONE ERROR PER TEST	
(1)	001612	012737	001612	MOV	#.,#LPADR	;;INITIALIZE THE LOOP ADDRESS FOR SCOPE	
(1)	001620	012737	001620	MOV	#.,#LPERR	;;SETUP THE ERROR LOOP ADDRESS	
(2)				;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS			
(2)				;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.			
(2)	001626	013746	000004	MOV	#ERRVEC,-(SP)	;;SAVE ERROR VECTOR	
(2)	001632	012737	001666	MOV	#648,#ERRVEC	;;SET UP ERROR VECTOR	
(2)	001640	012737	177570	MOV	#DSWR,SWR	;;SETUP FOR A HARDWARE SWICH REGISTER	
(2)	001646	012737	177570	MOV	#DISP,DISPLAY	;;AND A HARDWARE DISPLAY REGISTER	
(2)	001654	022777	177777	CMP	#-1,#SWR	;;TRY TO REFERENCE HARDWARE SWR	
(2)	001662	001012		BNE	668	;;BRANCH IF NO TIMEOUT TRAP OCCURRED	
(2)						;;AND THE HARDWARE SWR IS NOT "-1"	
(2)	001664	000403		BR	658	;;BRANCH IF NO TIMEOUT	
(2)	001666	012716	001674	MOV	#658,(SP)	;;SET UP FOR TRAP RETURN	
(2)	001672	000002		RTI			
(2)	001674	012737	000176	MOV	#SWREG,SWR	;;POINT TO SOFTWARE SWR	
(2)	001702	012737	000174	MOV	#DISPREG,DISPLAY	;;DISPLAY REGISTER	
(2)	001710	012637	000004	MOV	(SP)+,#ERRVEC	;;RESTORE ERROR VECTOR	
(1)							
(2)	001714	005037	001202	CLR	SPASS	;;CLEAR PASS COUNT	
(2)	001720	132737	000200	BITR	#APTSIZE,SENVH	;;TEST USER SIZE UNDER APT	
(2)	001726	001403		BEQ	678	;;YES,USE NON-APT SWITCH	
(2)	001730	012737	001216	MOV	#SWREG,SWR	;;NO,USE APT SWITCH REGISTER	
(2)	001736						
5697	001736	005037	007004	CLR	BADUNT	;;RESET BAD INDICATOR	
5698	001742	000137	002044	JMP	INIT1		

```

5700                                ;SUBROUTINE TO LOAD A TRAP CATCHER
5701
5702 001746 012702 000252 LDTRAP: MOV #252,R2 ;LOAD R2
5703 001752 012701 000250 MOV #250,R1 ;LOAD R1
5704 001756 010221 000250 50: MOV R2,(R1)+ ;LOAD .+2
5705 001760 005021 000250 CLR (R1)+ ;LOAD HALT
5706 001762 010122 000250 MOV R1,R2 ;LOAD R2
5707 001764 005722 000250 TST (R2)+ ;BUMP R2
5708 001766 020227 001002 CMP R2,#1002 ;TEST FOR LAST
5709 001772 001371 000250 BNE 55 ;BR UNTIL DONE
5710
5711                                ;AND LOAD DEVICE ADDRESSES LOCATIONS
5712
5713 001774 013700 001250 MOV #BASE,R0 ;GET BASE ADDRESS
5714 002000 010037 001422 MOV R0,DAC0 ;LOAD X ADDRESS
5715 002004 010037 001424 MOV R0,DAC1 ;LOAD Y ADDRESS
5716 002010 010037 001426 MOV R0,DAC2 ;LOAD DAC #2
5717 002014 010037 001430 MOV R0,DAC3 ;LOAD DAC #3
5718 002020 062737 000002 001424 ADD #2,DAC1
5719 002026 062737 000004 001426 ADD #4,DAC2
5720 002034 062737 000006 001430 ADD #6,DAC3
5721 002042 000207 000250 RTS ;EXIT
5722
5723 002044 004737 001746 INIT1: JSR PC,LDTRAP
5724 002050 005737 007012 TST TEMP ;TEST IF START OR RESTART
5725 002054 001012 000250 RNF MTFST ;RESTART
5726 002056 005737 000042 TST #42 ;TEST IF MONITOR
5727 002062 001007 000250 BNE MTFST ;BR IF NOT
5728 002064 005737 007020 TST WFTST ;TEST IF ON TESTER
5729 002070 001402 000250 BEQ 19 ;BR IF NOT
5730 002072 104401 000250 TYPE MSGSW ;TELL OPERATOR ABOUT TESTER SWITCHES
5731 002074 010011 000250 19: TYPE ;CALL MESSAGE PRINTER VIA 'EMT'
5732 002076 104401 000250 TITLE ;TYPE PROGRAM HEADER.
5733 002100 007040 000250
    
```

```

5741                                ;SRTTL DETERMINE THE NUMBER OF AAV11 ON THIS SYSTEM
5742
5743 002102 013737 001250 001126 MTEST: MOV #BASE,#RODAT ;GET THE BASE ADDRESS
5744 002110 005037 007010 CLR MASKNM ;CLEAR UNIT #
5745 002114 005037 001206 CLR #UNIT ;LOAD TRAP RETURN
5746 002120 012737 002164 000004 19: MOV #25,ERRVEC ;TEST IF ADDR EXISTS
5747 002126 005777 176774 TST #RODAT ;UPDATE THE BUS ADDRESS
5748 002132 063737 001416 001126 ADD VADDR,#RODAT ;UPDATE UNIT COUNT
5749 002140 005237 001206 INC #UNIT ;TEST IF "NO NOT SIZE"
5750 002144 005737 001214 TST #ENV ;BR IF NO SIZING
5751 002150 100413 000250 BMI 38 ;TEST IF MAX. NUMBER
5752 002152 023737 007006 001206 CMP NUMROK,#UNIT ;BR IF NOT
5753 002160 001362 000250 BNE 19 ;BR IF MAX.
5754 002162 000406 000250 BR 38 ;BR IF MAX.
5755 002164 022626 000250 29: CMP (SP)+,(SP)+ ;CLEAN THE STACK
5756 002166 005737 001206 TST #UNIT ;TEST IF ANY EXIST
5757 002172 001002 000250 RNF 38 ;BR IF SOME ARE THERE
5758 002174 104001 000250 ERROR 1 ;BASE ADDRESS CAUSED AN BUS TRAP
5759 002176 000443 000250 BR TST1 ;IS SBASE CORRECT??
5760 002200 005737 001420 38: TST EVFR ;TEST IF # HAS BEEN REPORTED
5761 002204 100422 000250 BMI 48 ;BR IF IT WAS
5762 002206 005737 007020 TST WFTST ;TEST IF TESTER NONE
5763 002212 001010 000250 BNE 68 ;BR IF TESTER
5764 002214 104401 000250 TYPE ;TELL OPERATOR THE # OF AAV11'S
5765 002216 010224 000250 FOUND1 MOV #UNIT,-(SP)
5766 002220 013746 001206 TYPOS ;
5767 002224 104403 000250 .BYTE 2
5768 002226 002 000250 .BYTE 0
5769 002227 000 000250 TYPE
5770 002230 104401 000250 FOUND2
5771 002232 010250 000250
5772 002234 013737 001206 001420 68: MOV #UNIT,EVER ;SAVE THE # OF AAV11'S FOR LATER
5773 002242 052737 100000 001420 BIS #RT15,EVER ;SET "REPORTED # FLAG"
5774 002250 000405 000250 BR ;
5775 002252 123737 001420 001206 48: CMPB EVER,#UNIT ;TEST IF ANY HAVE GONE AWAY
5776 002260 001401 000250 RMB 58 ;BR IF ALL ARE STILL HERE
5777 002262 104013 000250 ERROR 19 ;EXISTING UNIT FAILED TO RESPOND NOW
5778 002264 005037 001206 58: CLR #UNIT ;RESET UNIT POINTER
5779 002270 005037 001204 CLR #DEVCT ;MAKE APT HAPPY
5780 002274 004737 001746 JSR PC,LDTRAP ;LOAD TRAP CATCHER AND BUS ADDRESSES
5781 002300 012737 000001 007010 MOV #BT0,MASKNM ;LOAD MASK NUMBER IF ERROR
    
```

```

5783 (3) (3) (2) 002306 000004
5784 002310 012737 002340 000004
5785 002316 005777 177100
5786 002322 005777 177076
5787 002326 005777 177074
5788 002332 005777 177072
5789 002336 000407
5790 002340 022626
5791 002342 104001
5792 002344 012737 000006 000004
5793 002352 000137 004530
5794 002356 012737 000006 000004
5795 (3) (3) (2) 002364 000004
5796 002366 005037 001124
5797 002372 013777 001124 177022
5798 002400 017737 177016 001126
5799 002406 023737 001124 001126
5800 002414 001401
5801 002416 104002
5802
5803 (3) (3) (2) 002420 000004
5804 002422 012737 007777 001124
5805 002430 013777 001124 176764
5806 002436 017737 176760 001126
5807 002444 023737 001124 001126
5808 002452 001401
5809 002454 104002
5810
5811 (3) (3) (2) 002456 000004
5812 002460 012737 000100 001160
5813 002466 012737 004000 001124
5814 002474 013777 001124 176720
5815 002502 017737 176714 001126
5816 002510 023737 001124 001126
5817 002516 001401
5818 002520 104002
5819 002522 006237 001124
5820 002526 001362

//*****
//TEST 1 TEST THAT THE AAV11 RESPONDS TO THE CPU
//*****
TST1: SCOPE
MOV #19,ERRVEC ILOAD BUS TRAP RETURN
TST #DAC0 ITEST DAC #0
TST #DAC1 ITEST DAC #1
TST #DAC2 ITEST DAC #2
TST #DAC3 ITEST DAC #3
BR 25 IJBR AND RSTORE LOC. 4
1S: CMP (SP)+,(SP)+ ICLEAN THE STACK
ERROR 1 IERROR, BUS TIMEOUT WHEN ADDRESSING THE AAV11
MOV #6,ERRVEC ILOAD LOC 4
JMP REMAIN ITEST IF ANY OTHER'S
2S: MOV #6,ERRVEC ILOAD RETURN
//*****
//TEST 2 TEST THAT DAC0 REGISTER CAN BE CLEARED
//*****
TST2: SCOPE
CLR $GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC0 ILOAD REG
MOV #DAC0,$SDDAT IREAD REG
CMP $GDDAT,$SDDAT ICOMPARE
BEQ TST3 IJBR IF EQUAL
ERROR 2 IERROR, DAC0 REGISTER NOT = 0
//*****
//TEST 3 TEST THAT DAC0 REGISTER CAN BE LOADED WITH #7777
//*****
TST3: SCOPE
MOV #7777,$GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC0 ILOAD REG
MOV #DAC0,$SDDAT IREAD REG
CMP $GDDAT,$SDDAT ICOMPARE
BEQ TST4 IJBR IF EQUAL
ERROR 2 IERROR, DAC0 REGISTER NOT = 7777
//*****
//TEST 4 TEST THAT DAC0 REGISTER CAN HOLD A FLOATING 1 PATTERN
//*****
TST4: SCOPE
MOV #100,$TIMES IJDO 100 ITERATIONS
MOV #BIT11,$GDDAT ILOAD EXPECTED
1S: MOV $GDDAT,#DAC0 ILOAD DAC0 REGISTER
MOV #DAC0,$SDDAT IREAD THE REGISTER
CMP $GDDAT,$SDDAT ICOMPARE THE DATA
BEQ 25 IJBR IF SAME
ERROR 2 IERROR, DAC0 REGISTER FAILED TO HOLD A FLOATING
ASR $GDDAT ICHANGE THE DATA
RNE 1S IJBR AND TEST MORE DATA

```

```

5821 (4) (4) (3) 002530 000004
5822 002532 012737 000100 001160
5823 002540 012737 004000 001124
5824 002546 013777 001124 176646
5825 002554 017737 176642 001126
5826 002554 023737 001124 001126
5827 002562 023737 001124 001126
5828 002570 001401
5829 002572 017737 176624 001126
5830 002572 104002
5831 002600 001401
5832 002602 013777 001124 176612
5833 002610 006237 176606
5834 002614 006237 001124
5835 002620 001355
5836 (4) (4) (3) 002622 000004
5837 002624 012737 000010 001160
5838 002632 012737 007777 001124
5839 002640 013777 001124 176554
5840 002646 162737 000001 001124
5841 002654 162737 000001 176540
5842 002662 017737 176534 001126
5843 002670 023737 001124 001126
5844 002676 001401
5845 002700 104002
5846 002702 013777 001124 176512
5847 002710 005737 001124
5848 002714 001354
5849 (3) (3) (2) 002716 000004
5850 002720 005037 001124
5851 002724 013777 001124 176472
5852 002732 017737 176466 001126
5853 002740 023737 001124 001126
5854 002746 001401
5855 002750 104003
5856 (3) (3) (2) 002752 000004
5857 002754 012737 007777 001124
5858 002762 013777 001124 176434
5859 002770 017737 176430 001126
5860 002776 023737 001124 001126
5861 003004 001401
5862 003006 104003

//*****
//TEST 5 TEST THAT DAC0 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
//*****
TST5: SCOPE
MOV #100,$TIMES IJDO 100 ITERATIONS
MOV #BIT11,$GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC0 ILOAD DAC0
MOV #DAC0,$SDDAT IREAD THE REGISTER
CMP $GDDAT,$SDDAT ICOMPARE THE GOOD TO DAC0
BEQ 25 IJBR IF THE SAME
MOV #DAC0,$SDDAT ISAVE FOR TYPEOUT
ERROR 2 IDAC0 FAILED TO HOLD A FLOATING 1 PATTERN
MOV $GDDAT,#DAC0 ILOAD DAC0 AGAIN
ASR #DAC0 ICHANGE THE DATA
ASR $GDDAT ICHANGE THE EXPECTED
RNE 1S IJBR IF MORE DATA
//*****
//TEST 6 TEST THE "SUB" INSTRUCTION WORKS ON DAC0
//*****
TST6: SCOPE
MOV #10,$TIMES IJDO 10 ITERATIONS
MOV #7777,$GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC0 ILOAD DAC0
1S: SUB #1,$GDDAT ISUB A VALUE
SUB #1,#DAC0 IFROM EXPECTED AND DAC0
MOV #DAC0,$SDDAT IREAD THE REGISTER
CMP $GDDAT,$SDDAT ICOMPARE
BEQ 25 IJBR IF SAME
ERROR 2 ITHE SUB INSTRUCTION FAILED ON DAC0
MOV $GDDAT,#DAC0 ILOAD THE REGISTER AGAIN
TST $GDDAT ITEST FOR MORE DATA
2S: TST $GDDAT
RNE 1S IJBR IF MORE DATA
//*****
//TEST 7 TEST THAT DAC1 REGISTER CAN BE CLEARED
//*****
TST7: SCOPE
CLR $GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC1 ILOAD DAC1
MOV #DAC1,$SDDAT IREAD REG
CMP $GDDAT,$SDDAT ICOMPARE
BEQ TST10 IJBR IF EQUAL
ERROR 3 IERROR, DAC1 REGISTER NOT = 0
//*****
//TEST 10 TEST THAT DAC #1 REGISTER CAN BE LOADED WITH #7777
//*****
TST10: SCOPE
MOV #7777,$GDDAT ILOAD EXPECTED
MOV $GDDAT,#DAC1 ILOAD DAC #1
MOV #DAC1,$SDDAT IREAD REG
CMP $GDDAT,$SDDAT ICOMPARE
BEQ TST11 IJBR IF EQUAL
ERROR 3 IERROR, DAC1 REGISTER NOT = 7777

```

```

5838
(3)
(3)
(2) 003010 000004
(1) 003012 012737 000100 001160
5839 003020 012737 004000 001124
5840 003026 013777 001124 176370
5841 003034 017737 176364 001126
5842 003042 023737 001124 001126
5843 003050 001401
5844 003052 104003
5845 003054 006237 001124
5846 003060 001362

*****
)TEST 11 TEST THAT DAC #1 REGISTER CAN HOLD A FLOATING 1 PATTERN
*****
TST11: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #0111,SGDDAT //LOAD EXPECTED
101 MOV SGDDAT,#DAC1 //LOAD THE REGISTER
MOV #DAC1,SRODAT //READ THE REGISTER
CMP SGDDAT,SRODAT //COMPARE THE DATA
BEQ 20 //BR IF DATA IS SAME
ERROR 3 //ERROR, DAC #1 REGISTER FAILED TO HOLD A FLOATIN
201 ASR SGDDAT //CHANGE THE DATA
BNE 15 //RR AND TEST MORE DATA

*****
)TEST 12 TEST THAT DAC1 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
*****
TST12: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #0111,SGDDAT //LOAD EXPECTED
(1) 003100 013777 001124 176316
(1) 003106 017737 176312 001126
101 MOV #DAC1,SRODAT //LOAD DAC1
CMP SGDDAT,SRODAT //COMPARE THE GOOD TO DAC1
BEQ 20 //BR IF THE SAME
MOV #DAC1,SRODAT //SAVE FOR TYPEOUT
ERROR 3 //DAC1 FAILED TO HOLD A FLOATING 1 PATTERN
(1) 003124 017737 176274 001126
(1) 003132 104003
(1) 003134 013777 001124 176262
201 ASR #DAC1 //LOAD DAC1 AGAIN
ASR SGDDAT //CHANGE THE DATA
BNE 15 //CHANGE THE EXPECTED
//RR IF MORE DATA

*****
)TEST 13 TEST THE "SUR" INSTRUCTION WORKS ON DAC1
*****
TST13: SCOPE
MOV #10,STIMES //DO 10 ITERATIONS
MOV #7777,SGDDAT //LOAD EXPECTED
(1) 003172 013777 001124 176224
(1) 003200 162737 000001 001124
101 SUR #1,#DAC1 //SUR A VALUE
MOV #DAC1,SRODAT //FROM EXPECTED AND DAC1
CMP SGDDAT,SRODAT //READ THE REGISTER
BEQ 20 //COMPARE
(1) 003222 023737 001124 001126
(2) 003230 001404
(1) 003232 104003
(1) 003234 013777 001124 176162
201 TST SGDDAT,#DAC1 //THE SUR INSTRUCTION FAILED ON DAC1
TST SGDDAT //LOAD THE REGISTER AGAIN
BNE 15 //TEST FOR MORE DATA
//RR IF MORE DATA

```

```

5850
(3)
(3)
(2) 003250 000004
5851 003252 005037 001124
5852 003256 013777 001124 176142
5853 003264 017737 176136 001126
5854 003272 023737 001124 001126
5855 003300 001401
5856 003302 104004
5857
5858
(3)
(3)
(2) 003304 000004
5859 003306 012737 007777 001124
5860 003314 013777 001124 176104
5861 003322 017737 176100 001126
5862 003330 023737 001124 001126
5863 003336 001401
5864 003340 104004
5865
5866
(3)
(3)
(2) 003342 000004
(1) 003344 012737 000100 001160
5867 003352 012737 004000 001124
5868 003360 013777 001124 176000
5869 003366 017737 176034 001126
5870 003374 023737 001124 001126
5871 003400 001401
5872 003404 104004
5873 003406 006237 001124
5874 003412 001362

*****
)TEST 14 TEST THAT THE DAC #2 REGISTER CAN BE CLEARED
*****
TST14: SCOPE
CLP SGDDAT //LOAD EXPECTED
MOV #0000,SGDDAT //LOAD REG
MOV #DAC2,SRODAT //READ REG
CMP SGDDAT,SRODAT //COMPARE
BEQ 20 //BR IF EQUAL
ERROR 4 //ERROR, DAC #2 REGISTER NOT = 0

*****
)TEST 15 TEST THAT THE DAC #2 REGISTER CAN BE LOADED WITH #7777
*****
TST15: SCOPE
MOV #7777,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC2 //LOAD REG
MOV #DAC2,SRODAT //READ REG
CMP SGDDAT,SRODAT //COMPARE
BEQ 20 //BR IF EQUAL
ERROR 4 //ERROR, DAC #2 REGISTER NOT = 7777

*****
)TEST 16 TEST THAT THE DAC #2 REGISTER CAN HOLD A FLOATING 1 PATTERN
*****
TST16: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #0111,SGDDAT //LOAD EXPECTED
101 MOV SGDDAT,#DAC2 //LOAD DAC2 REGISTER
MOV #DAC2,SRODAT //READ THE REGISTER
CMP SGDDAT,SRODAT //COMPARE THE DATA
BEQ 20 //BR IF SAME
ERROR 4 //ERROR, DAC #2 REGISTER FAILED TO HOLD A FLOATIN
201 ASR SGDDAT //CHANGE THE DATA
BNE 15 //RR AND TEST MORE DATA

*****
)TEST 17 TEST THAT DAC2 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
*****
TST17: SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #0111,SGDDAT //LOAD EXPECTED
(1) 003416 012737 000100 001160
(1) 003424 012737 004000 001124
(1) 003432 013777 001124 175766
(1) 003440 017737 175762 001126
101 MOV #DAC2,SRODAT //LOAD DAC2
CMP SGDDAT,SRODAT //COMPARE THE GOOD TO DAC2
BEQ 20 //BR IF THE SAME
MOV #DAC2,SRODAT //SAVE FOR TYPEOUT
ERROR 4 //DAC2 FAILED TO HOLD A FLOATING 1 PATTERN
(1) 003456 017737 175744 001126
(1) 003464 104004
(1) 003466 013777 001124 175732
201 ASR #DAC2 //LOAD DAC2 AGAIN
ASR SGDDAT //CHANGE THE DATA
BNE 15 //CHANGE THE EXPECTED
//RR IF MORE DATA

```

```

5870 //*****
(4) J=TEST 20 TEST THE "SUB" INSTRUCTION WORKS ON DAC2
(4) //*****
(3) TST20: SCOPE
(2) 003506 000004 MOV #10,STIMES //DO 10 ITERATIONS
(1) 003510 012737 000010 001160 MOV #7777,SGDDAT //LOAD EXPECTED
(1) 003516 012737 007777 001124 MOV SGDDAT,#DAC2 //LOAD DAC2
(1) 003524 017777 001124 175674 18: SUR #1,SGDDAT //SUB A VALUE
(1) 003532 162737 000001 001124 SUR #1,#DAC2 //FROM EXPECTED AND DAC2
(1) 003540 162777 000001 175660 MOV #DAC2,SRDDAT //READ THE REGISTER
(1) 003546 017737 175654 001126 CMP SGDDAT,SRDDAT //COMPARE
(1) 003554 023737 001124 001126 BEQ 2# //BR IF SAME
(2) 003562 001404 ERROR 4 //THE SUR INSTRUCTION FAILED ON DAC2
(1) 003564 104004 MOV SGDDAT,#DAC2 //LOAD THE REGISTER AGAIN
(1) 003566 013777 001124 175632 23: TST SGDDAT //TEST FOR MORE DATA
(1) 003574 005737 001124 BNE 1# //BR IF MORE DATA
(2) 003600 001354

5879 //*****
5880 J=TEST 21 TEST THAT THE DAC #3 REGISTER CAN BE CLEARED
(3) //*****
(3) TST21: SCOPE
(2) 003602 000004 CLR SGDDAT //LOAD EXPECTED
5881 003604 005037 001124 MOV SGDDAT,#DAC3 //LOAD REG
5882 003610 013777 001124 175612 MOV #DAC3,SRDDAT //READ REG
5883 003616 017737 175606 001126 CMP SGDDAT,SRDDAT //COMPARE
5884 003624 023737 001124 001126 BEQ 5 //BR IF EQUAL
5885 003632 001401 ERROR 5 //ERROR, DAC #3 REGISTER NOT = 0
5886 003634 104005

5887 //*****
(3) J=TEST 22 TEST THAT THE DAC #3 REGISTER CAN BE LOADED WITH #7777
(3) //*****
(2) TST22: SCOPE
5889 003640 012737 007777 001124 MOV #7777,SGDDAT //LOAD EXPECTED
5890 003646 013777 001124 175554 MOV SGDDAT,#DAC3 //LOAD REG
5891 003654 017737 175550 001126 MOV #DAC3,SRDDAT //READ REG
5892 003662 023737 001124 001126 CMP SGDDAT,SRDDAT //COMPARE
5893 003670 001401 BEQ TST23 //BR IF EQUAL
5894 003672 104005 ERROR 5 //ERROR, DAC #3 REGISTER NOT = 7777
5895

5896 //*****
(3) J=TEST 23 TEST THAT THE DAC #3 REGISTER CAN HOLD A FLOATING 1 PATTERN
(3) //*****
(2) TST23: SCOPE
(1) 003676 012737 000100 001160 MOV #100,STIMES //DO 100 ITERATIONS
5897 003704 012737 004000 001124 MOV #BIT11,SGDDAT //LOAD EXPECTED
5898 003712 013777 001124 175510 18: MOV SGDDAT,#DAC3 //LOAD DAC #3 REGISTER
5899 003720 017737 175504 001126 MOV #DAC3,SRDDAT //READ THE REGISTER
5900 003726 023737 001124 001126 CMP SGDDAT,SRDDAT //COMPARE THE DATA
5901 003734 001401 BEQ 2# //BR IF SAME
5902 003736 104005 ERROR 5 //ERROR, DAC #3 REGISTER FAILED TO HOLD A FLOATIN
5903 003740 006237 001124 28: ASR SGDDAT //CHANGE THE DATA
5904 003744 001362 BNE 1# //BR AND TEST MORE DATA
5905

```

```

5907 //*****
(4) J=TEST 24 TEST THAT DAC3 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
(4) //*****
(3) TST24: SCOPE
(2) 003746 000004 MOV #100,STIMES //DO 100 ITERATIONS
(1) 003750 012737 000100 001160 MOV #BIT11,SGDDAT //LOAD EXPECTED
(1) 003756 012737 004000 001124 MOV SGDDAT,#DAC3 //LOAD DAC3
(1) 003764 013777 001124 175436 18: MOV #DAC3,SRDDAT //READ THE REGISTER
(1) 003772 017737 175432 001126 CMP SGDDAT,SRDDAT //COMPARE THE GOOD TO DAC3
(1) 004000 023737 001124 001126 BEQ 2# //BR IF THE SAME
(2) 004006 001407 MOV #DAC3,SRDDAT //SAVE FOR TYPEOUT
(1) 004010 017737 175414 001126 ERROR 5 //DAC3 FAILED TO HOLD A FLOATING 1 PATTERN
(1) 004016 104005 MOV SGDDAT,#DAC3 //LOAD DAC3 AGAIN
(1) 004020 013777 001124 175402 28: ASR #DAC3 //CHANGE THE DATA
(1) 004026 006277 175376 ASR SGDDAT //CHANGE THE EXPECTED
(1) 004032 006237 001124 BNE 1# //BR IF MORE DATA
(1) 004036 001355

5908 //*****
(4) J=TEST 25 TEST THE "SUB" INSTRUCTION WORKS ON DAC3
(4) //*****
(3) TST25: SCOPE
(2) 004040 000004 MOV #10,STIMES //DO 10 ITERATIONS
(1) 004044 012737 007777 001124 MOV #7777,SGDDAT //LOAD EXPECTED
(1) 004050 013777 001124 175344 MOV SGDDAT,#DAC3 //LOAD DAC3
(1) 004056 013777 001124 175344 18: SUR #1,SGDDAT //SUB A VALUE
(1) 004064 162737 000001 001124 SUR #1,#DAC3 //FROM EXPECTED AND DAC3
(1) 004072 162777 000001 175330 MOV #DAC3,SRDDAT //READ THE REGISTER
(1) 004100 017737 175324 001126 CMP SGDDAT,SRDDAT //COMPARE
(1) 004106 023737 001124 001126 BEQ 2# //BR IF SAME
(2) 004114 001404 ERROR 5 //THE SUB INSTRUCTION FAILED ON DAC3
(1) 004116 104005 MOV SGDDAT,#DAC3 //LOAD THE REGISTER AGAIN
(1) 004120 013777 001124 175302 28: TST SGDDAT //TEST FOR MORE DATA
(1) 004126 005737 001124 BNE 1# //BR IF MORE DATA
(2) 004132 001354

```

```

5910 J*****
(3) J=TEST 26 TEST THAT THE FOUR DAC REGISTERS CAN HOLD DIFFERENT DATA
(3) J*****
(2) TST26: SCOPE
5911 004134 000004 MOV #1111,0DAC0 ILOAD DAC #0
5912 004136 012777 001111 175256 MOV #2222,0DAC1 ILOAD DAC #1
5913 004144 012777 002222 175252 MOV #4444,0DAC2 ILOAD DAC #2
5914 004152 012777 004444 175246 MOV #7777,0DAC3 ILOAD DAC #3
5915 004160 012777 007777 175242 MOV #1111,$GDDAT ILOAD EXPECTED
5916 004166 012737 001111 001124 MOV #0AC0,$RDDAT IREAD RFG
5917 004174 017737 175222 001126 CMP $GDDAT,$RDDAT ICOMPARE
5918 004202 023737 001124 001126 BEQ 1# IJBR IF EQUAL
5919 004210 001401 ERROR 2 IERROR, SELECTED DAC #0 IN ERROR
5920
5921 004214 012737 002222 001124 1#1 MOV #2222,$GDDAT ILOAD EXPECTED
5922 004222 017737 175176 001126 MOV #0AC1,$RDDAT IREAD REG
5923 004230 023737 001124 001126 CMP $GDDAT,$RDDAT ICOMPARE
5924 004236 001401 RFG 2# IJBR IF EQUAL
5925 004240 104003 ERROR 3 IERROR, SELECTED DAC #1 IN ERROR
5926
5927 004242 012737 004444 001124 2#1 MOV #4444,$GDDAT ILOAD EXPECTED
5928 004250 017737 175152 001126 MOV #0AC2,$RDDAT IREAD REG
5929 004256 023737 001124 001126 CMP $GDDAT,$RDDAT ICOMPARE
5930 004264 001401 RFG 3# IJBR IF SAME
5931 004266 104004 ERROR 4 IERROR, SELECTED DAC #2 IN ERROR
5932
5933 004270 012737 007777 001124 3#1 MOV #7777,$GDDAT ILOAD EXPECTED
5934 004276 017737 175126 001126 MOV #0AC3,$RDDAT IREAD REG
5935 004304 023737 001124 001126 CMP $GDDAT,$RDDAT ICOMPARE
5936 004312 001401 RFG 4# IJBR IF SAME
5937 004314 104005 ERROR 5 IERROR, SELECTED DAC #3 IN ERROR

```

```

5939 J*****
(3) J=TEST 27 TEST THAT RESET CLEARS DAC #0 REGISTER
(3) J*****
(2) TST27: SCOPE
(1) 004316 000004 MOV #10,$TIMES IJDD 10 ITERATIONS
5940 004320 012737 000010 001160 MOV #-1,0DAC0 ILOAD EXPECTED
5941 004326 012777 177777 175066 CLR $GDDAT ILOAD EXPECTED
5942 004334 005037 001124 IREAD RFG
5943 004340 000005 RESET ICOMPARE
5944 004342 017737 175054 001126 MOV #0AC0,$RDDAT ICOMPARE
5945 004350 023737 001124 001126 CMP $GDDAT,$RDDAT ICOMPARE
5946 004356 001401 BEQ TST30 IJBR IF EQUAL
5947 004360 104002 ERROR 2 IERROR, RESET FAILED TO CLEAR DAC #0
5948
5949 J*****
(3) J=TEST 30 TEST THAT RESET CLEARS DAC #1 REGISTER
(3) J*****
(2) TST30: SCOPE
(1) 004364 000004 MOV #10,$TIMES IJDD 10 ITERATIONS
5949 004372 012777 177777 175024 MOV #-1,0DAC1 ILOAD EXPECTED
5950 004400 005037 001124 CLR $GDDAT IREAD RFG
5951 004404 000005 RESET ICOMPARE
5952 004406 017737 175012 001126 MOV #0AC1,$RDDAT ICOMPARE
5953 004414 023737 001124 001126 CMP $GDDAT,$RDDAT ICOMPARE
5954 004422 001401 BEQ TST31 IJBR IF EQUAL
5955 004424 104003 ERROR 3 IERROR, RESET FAILED TO CLEAR DAC #1
5956
5957 J*****
(3) J=TEST 31 TEST THAT RESET CLEARS DAC #2 REGISTER
(3) J*****
(2) TST31: SCOPE
(1) 004426 000004 MOV #10,$TIMES IJDD 10 ITERATIONS
5957 004430 012737 000010 001160 MOV #-1,0DAC2 ILOAD THE REGISTER
5958 004436 012777 177777 174762 CLR $GDDAT ILOAD THE EXPECTED
5959 004444 005037 001124 RESET ICOMPARE
5960 004446 017737 174750 001126 MOV #0AC2,$RDDAT IREAD THE REGISTER
5961 004452 017737 174750 001126 BEQ TST32 IJBR IF CLEARED
5962 004460 001401 ERROR 4 IERROR, RESET FAILED TO CLEAR DAC #2
5963
5964 J*****
(3) J=TEST 32 TEST THAT RESET CLEARS DAC #3 REGISTER
(3) J*****
(2) TST32: SCOPE
(1) 004464 000004 MOV #10,$TIMES IJDD 10 ITERATIONS
5964 004466 012737 000010 001160 MOV #-1,0DAC3 ILOAD THE REGISTER
5965 004474 012777 177777 174726 CLR $GDDAT ILOAD THE EXPECTED
5966 004502 005037 001124 MOV #1,TEMP RESET
5967 004506 012737 000001 007012 IREAD THE REGISTER
5968 004514 000005 BEQ TST33 IJBR IF CLEARED
5969 004516 017737 174706 001126 MOV #0AC3,$RDDAT IREAD THE REGISTER
5970 004524 001401 RFG 5 IJBR IF CLEARED
5971 004526 104005 ERROR 5 IERROR, RESET FAILED TO CLEAR DAC #3
5972
5973
5974

```

```

5976 004530          REMAINS:
5977                *****
(3)                ;=TEST 33      DETERMINE IF MORE AAV11'S REMAIN TO BE TESTED
(3)                *****
(2)                TST33:  SCOPE
(1) 004532 012737 000001 001160      MOV      #1,STIMES      ;IDO 1 ITERATION
5978 004540 005237 001206 001422      INC      SUNIT         ;UPDATE UNIT #
5979 004544 123737 001206 001420      CMPB    SUNIT,EVER    ;TEST IF MORE
5980 004552 001424                      REG      TST34        ;JBR IF NOT
5981 004554 005237 001204                      INC      SDEVCT       ;APT UNIT #
5982 004560 063737 001416 001422      ADD     VADDR,DAC0    ;UPDATE BUS ADDRESS
5983 004566 063737 001416 001424      ADD     VADDR,DAC1
5984 004574 063737 001416 001426      ADD     VADDR,DAC2
5985 004602 063737 001416 001430      ADD     VADDR,DAC3
5986 004610 006337 007010                      ASL     MASKNM        ;CHANGE THE ERROR FLAG BIT
5987 004614 005037 001102                      CLR     STBTHM
5988 004620 000137 002306                      JMP     TST1          ;TEST THE NEXT UNIT
5989
6015                *****
(3)                ;=TEST 34      DETERMINE IF RUNNING ON THE HARDWARE TESTER (IF NOT REPORT END OF PA
(3)                *****
(2)                TST34:  SCOPE
(1) 004624 000004                      MOV     #1,STIMES    ;IDO 1 ITERATION
6016 004634 005737 007020                      TST    WFTST         ;TEST IF ON TESTER
6017 004640 001002                      BNE    TST35        ;JBR TO TEST
6018 004642 000137 005454                      JMP     $ENDP

```

```

6020                *****
(3)                ;=TEST 35      TEST THAT DAC #3 OUTPUT BITS (0-3) FUNCTION
(3)                *****
(2) 004646 000004                      TST35:  SCOPE
(1) 004650 012737 000001 001160      MOV     #1,STIMES    ;IDO 1 ITERATION
6021 004656 012737 000010 007014      MOV     #R1T3,STEMP  ;LOAD DAC PATTERN
6022 004664 012737 004000 001124      MOV     #R1T11,SGDDAT ;LOAD EXPECTED PATTERN
6023
6024 004672 013777 007014 174530 1S1  MOV     STEMP,#DAC3   ;LOAD DAC REGISTER
6025 004700 017737 002120 001126      MOV     #ORIN,SGDDAT ;READ THE REGISTER
6026 004706 042737 170377 001126      BIC    #170377,SGDDAT ;MASK OFF OTHER BITS
6027 004714 023737 001124 001126      CMP    SGDDAT,SGDDAT ;COMPARE
6028 004722 001401                      BEQ    28            ;JBR IF THE SAME
6029 004724 104013                      ERROR  13           ;DAC #3 DIGITAL OUTPUT BITS IN ERROR
6030
6031 004726 006237 001124                      28:   ASR     SGDDAT    ;ADJUST EXPECTED
6032 004732 006237 007014                      ASR     STEMP       ;ADJUST LOADED PATTERN
6033 004736 001355                      BNE    18
6034
6035                *****
(3)                ;=TEST 36      VERIFY THE AAV11 +15 SUPPLY
(3)                *****
(2) 004740 000004                      TST36:  SCOPE
(1) 004742 012737 000001 001160      MOV     #1,STIMES    ;IDO 1 ITERATION
6036 004750 013737 007032 001124      MOV     V5744,SGDDAT ;LOAD EXPECTED
6037 004756 004537 006302                      JSR    RS,CONVRT     ;SAMPLE THE CHANNEL
6038 004762 000012                      12
6039 004764 013737 007034 007016      MOV     V144,SPREAD  ;LOAD TOLERANCE
6040 004772 004737 006520                      JSR    PC,COMPAR    ;TEST IT
6041 004776 000401                      BR     TST37        ;JBR
6042 005000 104011                      ERROR  11           ;+15 VOLT SUPPLY IS WRONG
6043
6044                *****
(3)                ;=TEST 37      VERIFY THE AAV11 -15 SUPPLY
(3)                *****
(2) 005002 000004                      TST37:  SCOPE
(1) 005004 012737 000001 001160      MOV     #1,STIMES    ;IDO 1 ITERATION
6045 005012 013737 007036 001124      MOV     V2034,SGDDAT ;LOAD EXPECTED
6046 005020 004537 006302                      JSR    RS,CONVRT     ;SAMPLE THE CHANNEL
6047 005024 000011                      11
6048 005026 013737 007034 007016      MOV     V144,SPREAD  ;LOAD TOLERANCE
6049 005034 004737 006520                      JSR    PC,COMPAR    ;TEST IT
6050 005040 000401                      BR     TST40        ;JBR
6051 005042 104012                      ERROR  12           ;-15 VOLT SUPPLY IS WRONG
6052

```

```

0054      ))*****
(4)      ))*TEST 40      DAC0 OFFSET ADJUSTMENT
(4)      ))*****
(3)      005044 000004      TST40: SCOPE
(2)      005046 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005054 005737 001202      TST      SPASS      //TEST IF FIRST PASS
(3)      005060 001006      BNE      TST41      //BR IF NOT
(1)
(1)      005062 004537 005634      JSR      RS,OFFDAC      //LOAD AND EXECUTE DAC OFFSET ADJ.
(1)      005066 001422      DAC0      //DAC ADDRESS
(1)      005070 010724      SELDR      //TYPEOUT ADDRESS
(1)      005072 010552      ADJR46      //RES. TO ADJUST
(1)      005074 000013      13      //RESULT CHANNEL #
(1)
(5)
(4)      ))*****
(4)      ))*TEST 41      DAC0 GAIN ADJUSTMENT
(4)      ))*****
(3)      005076 000004      TST41: SCOPE
(2)      005100 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005106 005737 001202      TST      SPASS      //TEST IF FIRST PASS
(3)      005112 001005      BNE      TST42      //BR IF NOT
(1)
(1)      005114 004537 005764      JSR      RS,GAINDAC      //LOAD AND EXECUTE DAC GAIN ADJ.
(1)      005120 001422      DAC0      //DAC ADDRESS
(1)      005122 010376      ADJR34      //RES. TO ADJUST
(1)      005124 000013      13      //CHANNEL # FOR RESULTS
(1)
(5)
(4)      ))*****
(4)      ))*TEST 42      DAC0 CALIBRATION
(4)      ))*****
(3)      005126 000004      TST42: SCOPE
(2)      005130 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005136 004537 006104      JSR      RS,CALDAC      //LOAD AND EXECUTE CALIBRATION
(1)      005142 001422      DAC0      //DAC ADDRESS
(1)      005144 000013      13      //CHANNEL # FOR RESULTS
    
```

```

0056      ))*****
(4)      ))*TEST 43      DAC1 OFFSET ADJUSTMENT
(4)      ))*****
(3)      005146 000004      TST43: SCOPE
(2)      005150 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005156 005737 001202      TST      SPASS      //TEST IF FIRST PASS
(3)      005162 001006      BNE      TST44      //BR IF NOT
(1)
(1)      005164 004537 005634      JSR      RS,OFFDAC      //LOAD AND EXECUTE DAC OFFSET ADJ.
(1)      005170 001424      DAC1      //DAC ADDRESS
(1)      005172 010744      SELD1      //TYPEOUT ADDRESS
(1)      005174 010605      ADJR47      //RES. TO ADJUST
(1)      005176 000014      14      //RESULT CHANNEL #
(1)
(5)
(4)      ))*****
(4)      ))*TEST 44      DAC1 GAIN ADJUSTMENT
(4)      ))*****
(3)      005200 000004      TST44: SCOPE
(2)      005202 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005210 005737 001202      TST      SPASS      //TEST IF FIRST PASS
(3)      005214 001005      BNE      TST45      //BR IF NOT
(1)
(1)      005216 004537 005764      JSR      RS,GAINDAC      //LOAD AND EXECUTE DAC GAIN ADJ.
(1)      005222 001424      DAC1      //DAC ADDRESS
(1)      005224 010431      ADJR35      //RES. TO ADJUST
(1)      005226 000014      14      //CHANNEL # FOR RESULTS
(1)
(5)
(4)      ))*****
(4)      ))*TEST 45      DAC1 CALIBRATION
(4)      ))*****
(3)      005230 000004      TST45: SCOPE
(2)      005232 012737 000001 001160      MOV      #1,STIMES      //DO 1 ITERATION
(1)      005240 004537 006104      JSR      RS,CALDAC      //LOAD AND EXECUTE CALIBRATION
(1)      005244 001424      DAC1      //DAC ADDRESS
(1)      005246 000014      14      //CHANNEL # FOR RESULTS
    
```

```

6050
(4)
(4)
(3) 005250 000004
(2) 005252 012737 000001 001160
(1) 005260 005737 001202
(3) 005264 001006
(1)
(1) 005266 004537 005634
(1) 005272 001426
(1) 005274 010762
(1) 005276 010640
(1) 005300 000016
(1)
(5)
(4)
(4)
(3) 005302 000004
(2) 005304 012737 000001 001160
(1) 005312 005737 001202
(3) 005316 001006
(1)
(1) 005320 004537 005764
(1) 005324 001426
(1) 005326 010664
(1) 005330 000016
(1)
(5)
(4)
(4)
(3) 005332 000004
(2) 005334 012737 000001 001160
(1) 005342 004537 006104
(1) 005346 001426
(1) 005350 000016

//*****
//TEST 46 DAC2 OFFSET ADJUSTMENT
//*****
TST46: SCOPE
MOV #1,STIMES //DO 1 ITERATION
TST SPASS //TEST IF FIRST PASS
BNE TST47 //BR IF NOT

JSR RS,OFFDAC //LOAD AND EXECUTE DAC OFFSET ADJ.
DAC2 //DAC ADDRESS
SELD2 //TYPEOUT ADDRESS
ADJR48 //RES. TO ADJUST
16 //RESULT CHANNEL #

//*****
//TEST 47 DAC2 GAIN ADJUSTMENT
//*****
TST47: SCOPE
MOV #1,STIMES //DO 1 ITERATION
TST SPASS //TEST IF FIRST PASS
BNE TST50 //BR IF NOT

JSR RS,GAINDAC //LOAD AND EXECUTE DAC GAIN ADJ.
DAC2 //DAC ADDRESS
ADJR36 //RES. TO ADJUST
16 //CHANNEL # FOR RESULTS

//*****
//TEST 50 DAC2 CALIBRATION
//*****
TST50: SCOPE
MOV #1,STIMES //DO 1 ITERATION
JSR RS,CALDAC //LOAD AND EXECUTE CALIBRATION
DAC2 //DAC ADDRESS
16 //CHANNEL # FOR RESULTS

```

```

6060
(4)
(4)
(3) 005352 000004
(2) 005354 012737 000001 001160
(1) 005362 005737 001202
(3) 005366 001006
(1)
(1) 005370 004537 005634
(1) 005374 001430
(1) 005376 011000
(1) 005400 010673
(1) 005402 000015
(1)
(5)
(4)
(4)
(3) 005404 000004
(2) 005406 012737 000001 001160
(1) 005414 005737 001202
(3) 005420 001006
(1)
(1) 005422 004537 005764
(1) 005426 001430
(1) 005430 010517
(1) 005432 000015
(1)
(5)
(4)
(4)
(3) 005434 000004
(2) 005436 012737 000001 001160
(1) 005444 004537 006104
(1) 005450 001430
(1) 005452 000015

//*****
//TEST 51 DAC3 OFFSET ADJUSTMENT
//*****
TST51: SCOPE
MOV #1,STIMES //DO 1 ITERATION
TST SPASS //TEST IF FIRST PASS
BNE TST52 //BR IF NOT

JSR RS,OFFDAC //LOAD AND EXECUTE DAC OFFSET ADJ.
DAC3 //DAC ADDRESS
SELD3 //TYPEOUT ADDRESS
ADJR49 //RES. TO ADJUST
15 //RESULT CHANNEL #

//*****
//TEST 52 DAC3 GAIN ADJUSTMENT
//*****
TST52: SCOPE
MOV #1,STIMES //DO 1 ITERATION
TST SPASS //TEST IF FIRST PASS
BNE TST53 //BR IF NOT

JSR RS,GAINDAC //LOAD AND EXECUTE DAC GAIN ADJ.
DAC3 //DAC ADDRESS
ADJR37 //RES. TO ADJUST
15 //CHANNEL # FOR RESULTS

//*****
//TEST 53 DAC3 CALIBRATION
//*****
TST53: SCOPE
MOV #1,STIMES //DO 1 ITERATION
JSR RS,CALDAC //LOAD AND EXECUTE CALIBRATION
DAC3 //DAC ADDRESS
15 //CHANNEL # FOR RESULTS

```



```

        .SBTTL SUBROUTINE TO TEST THE D/A CALIBRATION
0120
0129
0130 006104 012537 006210 CALDAC: MOV (R5)+,108 IGET BUS ADDRESS
0131 006110 017737 000074 006210 MOV 0108,108 I
0132 006116 013737 006210 007002 MOV 108,DACBAD ILOAD BUS ADDRESS IF ERROR
0133 006124 012537 006150 MOV (R5)+,118 IGET CHANNEL #
0134
0135 006130 012777 007400 000052 MOV #7400,0108 ILOAD THE DAC
0136 006136 012737 007400 001124 MOV #7400,SGDDAT ILOAD THE EXPECTED VALUE
0137
0138 006144 004537 006302 10: JSR R5,CONVRT ISAMPLE THE CHANNEL
0139 006150 000013 11: 13
0140
0141 006152 012737 000003 007016 MOV #3,SPREAD ILOAD TOLERANCE
0142 006160 004737 006520 JSR PC,COMPAR ITEST THE RESULTS
0143 006164 000491 BR 29 IIBR
0144 006166 104010 ERROR 10 INON-LINEARITY IN DAC DETECTED
0145 006170 162777 000400 000012 2: SUR #400,0108 IADJUST THE CONTENTS
0146 006176 162737 000400 001124 SUR #400,SGDDAT IADJUST THE EXPECTED
0147 006204 001357 BNE 14 IIBR IF NOT DONE
0148 006206 000205 RTS R5 IEXIT
0149
0150 006210 000000 10: 0
0151
0152 .SBTTL SUBROUTINE TO LOAD A VOLTAGE INTO THE VOLTAGE SOURCE
0153
0154 006212 012500 SNOVLT: MOV (R5)+,R0 ILOAD THE POINTER
0155 006214 112001 2: MOV(R0)+,R1 IGET SOME DATA
0156 006216 001421 BR 38 IIBR IF TERM
0157 006220 110177 000576 MOV(R1),#FTLZ ILOAD THE DATA
0158 006224 012701 001000 MOV #1000,R1
0159 006230 005301 5: DEC R1 IDELAY
0160 006232 001376 BNF 54
0161 006234 052777 000200 000560 B7S #0177,#FILZ ISET BIT 7
0162 006242 012701 001000 MOV #1000,R1 ILOAD DELAY
0163 006246 005301 1: DEC R1 IDELAY
0164 006250 001376 BNF 14
0165 006252 042777 000200 000502 BIC #0177,#FILZ
0166 006260 000745 RR 24
0167
0168 006262 012701 000000 3: MOV #0,R1 ILOAD DELAY
0169 006266 152777 000177 000526 B7SR #177,#FILZ IDISABLE BITS
0170 006274 005301 4: DEC R1 IDELAY
0171 006276 001376 BNF 64
0172 006300 000205 RTS R5 IEXIT
0173
    
```

```

        .SBTTL SUBROUTINE TO CONVERT CHANNEL N ON THE TESTER A/D
0175
0176
0177 006302 012537 006414 CONVRT: MOV (R5)+,108 IGET THE CHANNEL #
0178 006306 000337 006414 SWAB 108
0179 006312 042737 170377 006414 FIC #170377,108 IMASK OUT OTHER BITS
0180 006320 013777 006414 000500 MOV 108,#ADCS ISELECT CHANNEL
0181 006324 005037 006416 CLR 11 I
0182 006332 012737 000200 006420 MOV #0177,128 ILOAD SHIFT COUNTER
0183 006340 105277 000462 1: INCR #ADCS ICONVERT CHANNEL
0184 006344 105777 000456 2: TSTB #ADCS IWAIT FOR DONE
0185 006350 100375 RPL 24
0186 006352 067737 000452 006416 ADD #0000,118 IUPDATE CONVERSION
0187 006360 006237 006420 ASR 128 IFINISHED ?
0188 006364 001345 BNF 13
0189 006366 000257 CCC
0190 006370 006037 006416 ROR 118
0191 006374 006237 006416 ASR 118
0192 006400 006237 006416 ASR 118 IJUSTIFY DATA
0193 006404 013737 006416 001126 MOV 118,SHDDAT ILOAD ACTUAL <ADJUSTED>
0194 006412 000205 RTS R5 IEXIT
0195
0196 006414 000000 10: 0
0197 006416 000000 11: 0
0198 006420 000000 12: 0
0199
0200 .SBTTL SUBROUTINE TO LOOP UNTIL OPERATOR TYPES AN "SPACE"
0201
0202 006422 104401 007746 CSPACE: TYPE, LOSPAC ITELL OPERATOR TO HIT SPACE BAR
0203 006426 012737 000014 006516 MOV #10,118 ILOAD DELAY COUNTER
0204 006430 005037 006514 CLR 108
0205 006440 105777 172500 1: TSTB #STKS IWAIT FOR OPERATOR
0206 006444 100410 BMT 29 IIBR IF FLAG IS SET
0207 006446 005337 006514 DEC 108 IDELAY
0208 006452 001372 BNE 14 IIBR IF NOT DONE
0209 006456 005337 006516 DEC 118 IDELAY AGAIN
0210 006460 001367 BNF 14
0211 006462 104014 ERROR 14 IWAKE UP OPERATOR
0212 006464 000740 BR 38 ILOOP
0213 006466 017737 172454 006514 2: MOV #STKB,108 IREAD THE CHARACTER
0214 006474 042737 177600 006514 BIC #177600,108 IMASK OF OTHER BITS
0215 006502 022737 000040 006514 CMP #40,108 ITEST FOR "SPACE"
0216 006510 001346 BNE 34 ILOOP
0217 006512 000207 RTS PC IEXIT
0218 006514 000000 10: 0
0219 006516 000010 11: RTS
0220
    
```

```

        6222                .SBTTL  SUBROUTINE TO COMPARE TWO LOCATIONS BY THE SPREAD
        6223
        6224 006520 010046          COMPARI  MOV  R0,=(SP)           JSAVE R0
        6225 006522 010146          MOV  R1,=(SP)           JSAVE R1
        6226 006524 013700 001124  MOV  $GD0AT,R0        JGET EXPECTED VALUE
        6227 006530 013701 001126  MOV  $R0DAT,R1        JGET THE UNKNOWN
        6228 006534 160100          SUR  R1,R0             JSUBTRACT
        6229 006536 100001          BPL  03
        6230 006540 005400          NEG  R0
        6231 006542 020037 007016  001  CMP  R0,SPREAD      JTFST IF DIFFERENCE IF > THAN SPREAD
        6232 006546 003405          BLE  105
        6233 006550 012601          901  MOV  (SP)+,R1      JRESTORE R1
        6234 006552 012600          MOV  (SP)+,R0        JRESTORE R0
        6235 006554 062716 000002  ADD  #2,(SP)         JMAKE AN ERROR EXIT
        6236 006560 000207          RTS  PC              JEXIT
        6237
        6238 006562 012601          105: MOV  (SP)+,R1
        6239 006564 012600          MOV  (SP)+,R0
        6240 006566 000207          RTS  PC              JEXIT FOR GOOD LIMIT TEST
        6241
        6242                .SBTTL  FULL SCALE RAMP ON EACH RAMP
        6243
        6244 006570 012706 001100  FULRMP: MOV  #STACK,SP      JLOAD POINTER
        6245 006574 004737 001746  JSR  PC,LDTRAP       JLOAD BUS ADDRESS
        6246 006600 013700 001422  101  JSR  DAC0,R0         JGET BUS ADDRESS
        6247 006604 004737 006642  JSR  PC,103          JLOAD THE RAMP ON DAC #1
        6248 006610 013700 001424  MOV  DAC1,R0         JGET BUS ADDRESS
        6249 006614 004737 006642  JSR  PC,103          JLOAD THE RAMP ON DAC #1
        6250 006620 013700 001426  MOV  DAC2,R0         JGET BUS ADDRESS
        6251 006624 004737 006642  JSR  PC,103          JLOAD THE RAMP ON DAC #2
        6252 006630 013700 001430  MOV  DAC3,R0         JGET THE BUS ADDRESS
        6253 006634 004737 006642  JSR  PC,103          JLOAD THE RAMP ON DAC #3
        6254 006640 000757          BR  15
        6255
        6256 006642 005010          105: CLR  (R0)         JCLEAR DAC
        6257 006644 062710 000010  115: ADD  #10,(R0)      JUPDATE THE DATA
        6258 006650 005710          TST  (R0)           JTEST IF DONE
        6259 006652 001374          RNE  115            JBR IF NOT
        6260 006654 000207          RTS  PC              JEXIT
    
```

```

        6268                .SBTTL  STATIC DAC CALIBRATION
        6269
        6270 006656 012706 001100  STATIC: MOV  #STACK,SP      JLOAD STACK POINTER
        6271 006662 004737 001746  JSR  PC,LDTRAP       JLOAD BUS ADDRESSES
        6272 006666 104410          101  CKSWR          JTEST FOR CTRL G
        6273 006670 017700 172204  MOV  #SWR,R0         JREAD SWITCHES
        6274 006674 010077 172522  MOV  R0,#DAC0        JLOAD DAC #0
        6275 006678 010077 172520  MOV  R0,#DAC1        JLOAD DAC #1
        6276 006682 010077 172516  MOV  R0,#DAC2        JLOAD DAC #2
        6277 006686 010077 172514  MOV  R0,#DAC3        JLOAD DAC #3
        6278 006690 000764          RR  15
        6279
        6280                .SBTTL  DYNAMIC DAC CALIBRATION
        6281
        6282 006716 012706 001100  DYNCAL: MOV  #STACK,SP      JLOAD STACK POINTER
        6283 006722 004737 001746  JSR  PC,LDTRAP       JLOAD BUS ADDRESSES
        6284 006726 104410          101  CKSWR          JTEST FOR CTRL G
        6285 006730 017700 172204  MOV  #SWR,R0         JREAD SWR
        6286 006734 004737 006750  JSR  PC,103          JLOAD THE SWR VALUE TO ALL DACS
        6287 006740 005000          CLR  R0              JCLEAR R0
        6288 006742 004737 006750  JSR  PC,103          JLOAD ALL DAC'S WITH 0
        6289 006746 000767          BR  15
        6290
        6291 006750 010077 172446  105: MOV  R0,#DAC0        JLOAD DAC #0
        6292 006754 010077 172444  MOV  R0,#DAC1        JLOAD DAC #1
        6293 006760 010077 172442  MOV  R0,#DAC2        JLOAD DAC #2
        6294 006764 010077 172440  MOV  R0,#DAC3        JLOAD DAC #3
        6295 006770 012700 000020  MOV  #20,R0          JLOAD DELAY COUNTER
        6296 006774 005300          115: DEC  R0           JDELAY
        6297 006776 100376          BPL  115            JWAIT
        6298 007000 000207          RTS  PC              JEXIT
        6300
        6301 007002 170440          DACBAD: ABASE
        6302 007004 000000          BADUNT: 0
        6303 007006 000004          NUMBOOK: 4,
        6304 007010 000001          MASKNM: 010
        6305 007012 000000          TEMP: 0
        6306 007014 000000          STEMP: 0
        6307 007016 000000          SPREAD: 0
        6308 007020 000000          WFTST: 0
        6309 007022 167772          FILZ: 167772
        6310 007024 167774          DRINI: 167774
        6311 007026 170500          ADCS: 170500
        6312 007030 170502          ADDR: 170502
        6313 007032 005744          V5744: 5744
        6314 007034 000144          V144: 144
        6315 007036 002034          V2034: 2034
        6316
        6322
    
```

6324  
6325  
6326

.BRTTL .ASCII MESSAGES

6327 007040 005015 040412 053101 TITLE: .ASCII <15><12><12>'AAV11 DIAGNOSTIC TEST, (MAINDEC-11-DVAAA-AP)'<15><12>  
 027046 030461 042040 040511  
 007054 047107 051517 044524  
 007062 020103 042524 052123  
 007070 020054 046450 044501  
 007076 042116 041505 030455  
 007104 026461 053104 040501  
 007112 026501 030101 006451  
 007120 000012  
 6328 007122 052502 020123 044524 EM11 .ASCII /BUS TIME-OUT WHEN REFERENCING A DAC ADDRESS/  
 007130 042515 047455 052125  
 007136 053440 042510 020116  
 007144 042522 042506 042522  
 007152 041516 047111 020107  
 007160 020101 040504 020103  
 007166 042101 051104 051505  
 007174 000123  
 6329 007176 040504 030103 051040 EM21 .ASCII /DAC0 REGISTER IN ERROR/  
 007204 043505 051511 042524  
 007212 020122 047111 042440  
 007220 051122 051117 000  
 6330 007225 104 041501 020061 EM31 .ASCII /DAC1 REGISTER IN ERROR/  
 007232 042522 044507 052123  
 007240 051105 044440 020116  
 007246 051105 047522 000122  
 6331 007254 040504 031103 051040 EM41 .ASCII /DACR REGISTER IN ERROR/  
 007262 043505 051511 042524  
 007270 020122 047111 042440  
 007276 051122 051117 000  
 6332 007303 104 041501 020063 EM51 .ASCII /DAC3 REGISTER IN ERROR/  
 007310 042522 044507 052123  
 007316 051105 044440 020116  
 007324 051105 047522 000122  
 6333 007332 042523 042514 052103 EM61 .ASCII /SELECTED DAC OFFSET POT WAS ADJUSTED INCORRECTLY/  
 007340 042105 042040 041501  
 007346 047440 043106 042523  
 007354 020124 047520 020124  
 007362 040527 020123 042101  
 007370 052512 052123 042105  
 007376 044440 041516 051117  
 007404 042522 052103 054514  
 007412 000  
 6334 007413 123 046105 041505 EM71 .ASCII /SELECTED DAC GAIN POT WAS ADJUSTED INCORRECTLY/  
 007420 042524 020104 040504  
 007426 020103 040507 047111  
 007434 050040 052117 053440  
 007442 051501 040440 045104  
 007450 051525 042524 020104  
 007456 047111 047503 051122  
 007464 041505 046124 000131  
 6335 007472 042523 042514 052103 EM101 .ASCII /SELECTED DAC HAS A LINEARITY PROBLEM/  
 007500 042105 042040 041501  
 007506 040440 051501 040440

6336  
6337  
6338  
6339  
6340  
6341  
6342  
6343  
6344

.ASCII MESSAGES

007514 046040 047111 040505  
 007522 044522 044524 050040  
 007530 047522 046102 046505  
 007536 000  
 6336 007537 053 032461 053040 EM111 .ASCII /+15 VOLT SUPPLY IS INCORRECT/  
 007544 046117 020124 052523  
 007552 050120 054514 044440  
 007560 020123 047111 047503  
 007566 051122 041405 000124  
 6337 007574 030455 020065 047526 EM121 .ASCII /-15 VOLT SUPPLY IS INCORRECT/  
 007602 052114 051440 050125  
 007610 046120 020131 051511  
 007616 044440 041516 051117  
 007624 042522 052103 000  
 6338 007631 104 041501 021440 EM131 .ASCII /DAC #3 DIGITAL OUTPUT BITS IN ERROR/  
 007636 020063 044504 044507  
 007644 040524 020114 052517  
 007652 050124 052125 041040  
 007660 052111 020123 047111  
 007666 042440 051122 051117  
 007674 000  
 6339 007675 007 003407 040527 EM141 .ASCII <7><7><7>/WAKE UP OPERATOR AND ADJUST THE POT/<7><7>  
 007702 042513 052440 020120  
 007710 050117 051105 052101  
 007716 051117 040440 042116  
 007724 040440 045104 051525  
 007732 020124 044124 020105  
 007740 047520 003524 000007  
 6340 007746 042011 050105 042522 L08PAC1 .ASCII / DEPRESS THE "SPACE-BAR" WHEN DONE/  
 007754 051523 052040 042510  
 007762 021040 050123 041501  
 007770 026505 040502 021122  
 007776 053440 042510 020116  
 6341 010004 047504 042516 000 MSG8W1 .ASCII <7><15><12>/TESTER SW1-2 AND SW1-5 ONLY MUST BE ON/  
 010011 007 050515 042524  
 010016 052123 051105 051440  
 010024 030527 031055 040440  
 010032 042116 051440 030527  
 010040 032455 047440 046116  
 010046 020131 052515 052123  
 010054 041040 020105 047117  
 6342 010062 006407 051412 031127 .ASCII <7><15><12>/SW2-2 SW2-4 AND SW2-5 MUST BE ON/  
 010070 031055 051440 031127  
 010076 032055 040440 042116  
 010104 051440 031127 030455  
 010112 040440 051525 020124  
 010120 042502 047440 116  
 6343 010125 015 003412 047503 .ASCII <15><12><7>/CONNECT AAV11 TO J09 OF TESTER ONLY/<15><12>  
 010132 047116 041505 020124  
 010140 040501 030524 020061  
 010146 047524 045040 030440  
 010154 047440 020106 042524  
 010162 052123 051105 047440  
 010170 046116 006531 000012  
 6344 010176 021440 042440 051122 ERRTOT1 .ASCII / # ERRORS/  
 010204 051117 000123







```

(1) 012234 006300 ASL R0
(1) 012236 006300 ASL R0
(1) 012240 002700 001256 ADD #ERRTB,R0 ;FORM TABLE POINTER
(1) 012244 012037 012254 MOV (R0)+,2S ;PICKUP "ERROR MESSAGE" POINTER
(1) 012250 001404 BEQ 3# ;SKIP TYPEOUT IF NO POINTER
(1) 012252 104401 TYPE ;TYPE THE "ERROR MESSAGE"
(1) 012254 000000 2S: ,WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
(1) 012256 104401 001171 TYPE ,SCRLF ;"CARRIAGE RETURN" & "LINE FEED"
(1) 012262 012037 012272 3S: MOV (R0)+,4S ;PICKUP "DATA HEADER" POINTER
(1) 012266 001404 REQ 5# ;SKIP TYPEOUT IF 0
(1) 012270 104401 TYPE ;TYPE THE "DATA HEADER"
(1) 012272 000000 4S: ,WORD 0 ;"DATA HEADER" POINTER GOES HERE
(1) 012274 104401 001171 TYPE ,SCRLF ;"CARRIAGE RETURN" & "LINE FEED"
(1) 012300 011000 5S: MOV (R0),R0 ;PICKUP "DATA TABLE" POINTER
(1) 012302 001004 BNE 7# ;JGO TYPE THE DATA
(1) 012304 012600 6S: MOV (SP)+,R0 ;RESTORE R0
(1) 012306 104401 001171 TYPE ,SCRLF ;"CARRIAGE RETURN" & "LINE FEED"
(1) 012312 000207 RTS ;RETURN
(2) 012314 7S: MOV # (R0)+,-(SP) ;SAVE # (R0)+ FOR TYPEOUT
(2) 012316 104402 TYPOC ;JGO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 012320 005710 TST (R0) ;IS THERE ANOTHER NUMBER?
(1) 012322 001770 BEQ 6# ;BR IF NO
(1) 012324 104401 012332 TYPE ,## ;TYPE TWO(2) SPACES
(1) 012330 000771 BR 7# ;JLOOP
(1) 012332 020040 000 .ASCIZ / / ;TWO(2) SPACES
(1) 012336 102336 .EVEN
6394 .SBYTL POWER DOWN AND UP ROUTINES
(1)
(2)
(1) ;*****
(1) ;POWER DOWN ROUTINE
(1) 012336 012737 012502 000024 SPWRDN: MOV #11LLUP,#PWRVEC ;SET FOR FAST UP
(1) 012344 012737 000340 000026 MOV #340,#PWRVEC+2 ;PRIORITY
(1) 012352 010046 MOV R0,-(SP) ;PUSH R0 ON STACK
(1) 012354 010146 MOV R1,-(SP) ;PUSH R1 ON STACK
(1) 012356 010246 MOV R2,-(SP) ;PUSH R2 ON STACK
(1) 012360 010346 MOV R3,-(SP) ;PUSH R3 ON STACK
(1) 012362 010446 MOV R4,-(SP) ;PUSH R4 ON STACK
(1) 012364 010546 MOV R5,-(SP) ;PUSH R5 ON STACK
(1) 012366 017746 166546 MOV #SWR,-(SP) ;PUSH #SWR ON STACK
(1) 012372 010677 012506 MOV SP,#SAVR6 ;SAVE SP
(1) 012376 012737 012410 000024 MOV #SPWRUP,#PWRVEC ;SET UP VECTOR
(1) 012404 000000 HALT
(1) 012406 000776 BR -2 ;HANG UP
(2)
(1) ;*****
(1) ;POWER UP ROUTINE
(1) 012410 012737 012502 000024 SPWRUP: MOV #11LLUP,#PWRVEC ;SET FOR FAST DOWN
(1) 012416 013700 012506 MOV #SAVR6,SP ;GET SP
(1) 012422 005037 012506 CLR #SAVR6 ;WAIT LOOP FOR THE TTY
(1) 012426 005237 012506 1S: INC #SAVR6 ;WAIT FOR THE INC
(1) 012432 001375 BNE 1# ;IF WORD
(1) 012434 012677 166500 MOV (SP)+,#SWR ;POP STACK INTO #SWR
(1) 012440 012605 MOV (SP)+,R5 ;POP STACK INTO R5
(1) 012442 012604 MOV (SP)+,R4 ;POP STACK INTO R4
(1) 012444 012603 MOV (SP)+,R3 ;POP STACK INTO R3

```

```

(3) 012446 012602 MOV (SP)+,R2 ;POP STACK INTO R2
(3) 012450 012601 MOV (SP)+,R1 ;POP STACK INTO R1
(3) 012452 012600 MOV (SP)+,R0 ;POP STACK INTO R0
(1) 012454 012737 012336 000024 MOV #SPWRDN,#PWRVEC ;SET UP THE POWER DOWN VECTOR
(1) 012462 012737 000340 000026 MOV #340,#PWRVEC+2 ;PRIORITY
(1) 012470 104401 TYPE ;REPORT THE POWER FAILURE
(1) 012472 012510 SPWRMG: ,WORD PWRMSG ;POWER FAIL MESSAGE POINTER
(1) 012474 012716 MOV (PC)+,(SP) ;RESTART AT BEGIN
(1) 012476 001450 SPWRAD: ,WORD BEGIN ;RESTART ADDRESS
(1) 012500 000002 RTI
(1) 012502 000000 STILLUP: HALT ;THE POWER UP SEQUENCE WAS STARTED
(1) 012504 000776 BR -2 ;BEFORE THE POWER DOWN WAS COMPLETE
(1) 012506 000000 #SAVR6: 0 ;PUT THE SP HERE
6395 012510 005015 042522 052123 PWRMSG: .ASCIZ <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>
012516 051101 044524 043516
012524 040440 052106 051105
012532 040440 050740 053517
012540 051105 043040 044501
012546 052514 042522 005015
012554 000012
6396 .EVEN
6397 .SBYTL BINARY TO OCTAL (ASCII) AND TYPE
6398
(1)
(2)
(1) ;*****
(1) ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
(1) ;OCTAL (ASCII) NUMBER AND TYPE IT.
(1) ;#STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
(1)
(1) ;#CALL:
(1) ;# MOV NUM,-(SP) ;NUMBER TO BE TYPED
(1) ;# TYPOS ;CALL FOR TYPEOUT
(1) ;# .BYTE N ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
(1) ;# .BYTE M ;M=1 OR 0
(1) ;# ;
(1) ;# ; ;#TYPE LEADING ZEROS
(1) ;# ; ;#SUPPRESS LEADING ZEROS
(1) ;#
(1) ;#STYPOC---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
(1) ;#STYPOS OR STYPOC
(1) ;#CALL:
(1) ;# TYPOC NUM,-(SP) ;NUMBER TO BE TYPED
(1) ;# ; ;CALL FOR TYPEOUT
(1) ;#
(1) ;#STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
(1) ;#CALL:
(1) ;# MOV NUM,-(SP) ;NUMBER TO BE TYPED
(1) ;# TYPOC ;CALL FOR TYPEOUT
(1) ;#
(1) 012556 017646 000000 013001 STYPOS: MOV # (SP)-,(SP) ;PICKUP THE MODE
(1) 012562 116637 000001 MOVB 1(SP),#ZFILL ;LOAD ZERO FILL SWITCH
(1) 012570 112637 013003 MOVB (SP)+,#MODE+1 ;NUMBER OF DIGITS TO TYPE
(1) 012574 062716 000002 ADD #2,(SP) ;ADJUST RETURN ADDRESS
(1) 012600 000000 BR STYPOC
(1) 012602 112737 000001 013001 STYPOC: MOVB #1,#ZFILL ;SET THE ZERO FILL SWITCH
(1) 012610 112737 000006 013003 MOVB #6,#MODE+1 ;SET FOR SIX(6) DIGITS
(1) 012616 112737 000005 013000 STYPOC: MOVB #5,#OCTN ;SET THE ITERATION COUNT
(1) 012624 010346 MOV R3,-(SP) ;SAVE R3

```

```

DVAAA,P11 BINARY TO OCTAL (ASCII) AND TYPE
(1) 012626 010446 MOV R4,-(SP) //SAVE R4
(1) 012630 010546 MOV R5,-(SP) //SAVE R5
(1) 012632 113704 013003 MOV#R SOMODE+1,R4 //GET THE NUMBER OF DIGITS TO TYPE
(1) 012636 005404 NEG R4
(1) 012640 002704 000006 ADD #6,R4 //SUBTRACT IT FOR MAX. ALLOWED
(1) 012644 110437 MOV#R R4,SOMODE //SAVE IT FOR USE
(1) 012650 113704 013001 MOV#B $0FILL,R4 //GET THE ZERO FILL SWITCH
(1) 012654 016605 000012 MOV#R 12(SP),R5 //PICKUP THE INPUT NUMBER
(1) 012660 005003 CLR R3 //CLEAR THE OUTPUT WORD
(1) 012662 006105 10: ROL R5 //ROTATE MSB INTO "C"
(1) 012664 000404 BR 30 //GO DO MSB
(1) 012666 006105 20: ROL R5 //FORM THIS DIGIT
(1) 012670 006105 ROL R5
(1) 012672 006105 ROL R5
(1) 012674 010503 MOV R5,R3
(1) 012676 006103 30: ROL R3 //GET LSR OF THIS DIGIT
(1) 012700 105337 013002 DECB SOMODE //TYPE THIS DIGIT?
(1) 012704 100016 RPL Y0 //BR IF NO
(1) 012706 042703 177770 BIC #177770,R3 //GET RID OF JUNK
(1) 012712 001002 BNE 40 //TEST FOR 0
(1) 012714 005704 TST R4 //SUPPRESS THIS 0?
(1) 012716 001403 BEQ 50 //BR IF YES
(1) 012720 005204 40: INC R4 //DON'T SUPPRESS ANYMORE 0'S
(1) 012722 052703 000060 BIC #'0,R3 //MAKE THIS DIGIT ASCII
(1) 012726 052703 000040 50: BIC #'0,R3 //MAKE ASCII IF NOT ALREADY
(1) 012732 110337 MOV#R R5,R5 //SAVE FOR TYPING
(1) 012736 104401 012776 TYPE ,R5 //GO TYPE THIS DIGIT
(1) 012742 105337 013000 70: DECB $OCNT //COUNT BY 1
(1) 012746 003347 BGT 20 //BR IF MORE TO DO
(1) 012750 002402 BLY 60 //BR IF DONE
(1) 012752 005204 INC R4 //INSURE LAST DIGIT ISN'T A BLANK
(1) 012754 000744 BR 20 //GO DO THE LAST DIGIT
(1) 012756 012605 60: MOV (SP)+,R5 //RESTORE R5
(1) 012760 012604 MOV (SP)+,R4 //RESTORE R4
(1) 012762 012603 MOV (SP)+,R3 //RESTORE R3
(1) 012764 016666 000002 000004 MOV 2(SP),4(SP) //SET THE STACK FOR RETURNING
(1) 012772 012616 MOV (SP)+,(SP)
(1) 012774 000002 RTI //RETURN
(1) 012776 000 00: .BYTE 0 //STORAGE FOR ASCII DIGIT
(1) 012777 000 00: .BYTE 0 //TERMINATOR FOR TYPE ROUTINE
(1) 013000 000 00: $OCNT: .BYTE 0 //OCTAL DIGIT COUNTER
(1) 013001 000 00: $0FILL: .BYTE 0 //ZERO FILL SWITCH
(1) 013002 000000 00: $SOMODE: .WORD 0 //NUMBER OF DIGITS TO TYPE
(1) 013003 000000 00: $RPTL: .TYPE ROUTINE
(1)
(1)
(1) //*****
(1) //ROUTINE TO TYPE ASCII MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
(1) //THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
(1) //NOTE1: NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
(1) //NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
(1) //NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
(1) //
(1) //CALL:
(1) //1) USING A TRAP INSTRUCTION
(1) // TYPE ,MESADR //MESADR IS FIRST ADDRESS OF AN ASCII STRING
(1) //OR

```

```

DVAAA,P11 TYPE ROUTINE
(1)
(1)
(1)
(1)
(1) 013004 105737 001157 STYPE: TSTB STPFLG //IS THERE A TERMINAL?
(1) 013010 100002 BPL 10 //BR IF YES
(1) 013012 000000 HALT //HALT HERE IF NO TERMINAL
(1) 013014 000430 BR 30 //LEAVE R0
(1) 013016 010046 10: MOV R0,-(SP) //SAVE R0
(1) 013020 017600 000002 001214 MOV #2(SP),R0 //GET ADDRESS OF ASCII STRING
(1) 013024 122737 000001 CMPB #4PTENV,SENV //RUNNING IN APT MODE
(1) 013032 001011 BNE 620 //NO,GO CHECK FOR APT CONSOLE
(1) 013034 132737 000100 001215 BIT# #4PTSPool,SENV //SPool MESSAGE TO APT
(1) 013042 001405 BEQ 620 //NO,GO CHECK FOR CONSOLE
(1) 013044 010037 013054 MOV R0,610 //SETUP MESSAGE ADDRESS FOR APT
(1) 013050 004737 014152 JSR #ATY3 //SPool MESSAGE TO APT
(1) 013054 000000 .WORD 0 //MESSAGE ADDRESS
(1) 013056 132737 000040 001215 BIT# #4PTCSUP,SENV //APT CONSOLE SUPPRESSED
(1) 013060 001003 BNE 600 //YES,SKIP TYPE OUT
(1) 013066 112046 20: MOV#R (R0)+,-(SP) //PUSH CHARACTER TO BE TYPED ONTO STACK
(1) 013070 001005 BNE 40 //BR IF IT ISN'T THE TERMINATOR
(1) 013072 005726 TST (SP)+ //IF TERMINATOR POP IT OFF THE STACK
(1) 013074 012600 MOV (SP)+,R0 //RESTORE R0
(1) 013076 062716 000002 30: ADD #2,(SP) //ADJUST RETURN PC
(1) 013102 000002 RTI //RETURN
(1) 013104 122716 000011 40: CMPB #HT,(SP) //BRANCH IF <HT>
(1) 013110 001430 BEQ 00 //
(1) 013112 122716 000200 CMPB #CRLF,(SP) //BRANCH IF NOT <CRLF>
(1) 013116 001006 BNE 50 //
(1) 013120 005726 TST (SP)+ //POP <CR><LF> EQUIV
(1) 013122 104401 TYPE //TYPE A CR AND LF
(1) 013124 001171 SCALF //
(1) 013126 105037 013262 CLRB $CHARCNT //CLEAR CHARACTER COUNT
(1) 013132 000755 BR 20 //GET NEXT CHARACTER
(1) 013134 004737 013216 50: JSR PC,STYPEC //GO TYPE THIS CHARACTER
(1) 013140 123726 001156 60: CMPB $FILLC,(SP)+ //IS IT TIME FOR FILLER CHARS.?
(1) 013144 001350 RNE 20 //IF NO GO GET NEXT CHAR.
(1) 013146 013746 001154 MOV #NULL,-(SP) //GET # OF FILLER CHARS. NEEDED
(1) //AND THE NULL CHAR.
(1) 013152 105366 000001 70: DECB 1(SP) //DOES A NULL NEED TO BE TYPED?
(1) 013156 002770 BLY 60 //BR IF NO=GO POP THE NULL OFF OF STACK
(1) 013160 004737 013216 JSR PC,STYPEC //GO TYPE A NULL
(1) 013164 105337 013262 DECB $CHARCNT //DO NOT COUNT AS A COUNT
(1) 013170 000770 BR 70 //LOOP
(1)
(1) //HORIZONTAL TAB PROCESSOR
(1)
(1) 013172 112716 000040 80: MOV#R #0,(SP) //REPLACE TAB WITH SPACE
(1) 013176 004737 013216 90: JSR PC,STYPEC //TYPE A SPACE
(1) 013202 132737 000007 013262 BIT# #7,$CHARCNT //BRANCH IF NOT AT
(1) 013210 001372 BNE 90 //TAB STOP
(1) 013212 005726 TST (SP)+ //POP SPACE OFF STACK
(1) 013214 000724 BR 20 //GET NEXT CHARACTER
(1) 013216 105777 165726 STYPECI: TSTB #0TB //WAIT UNTIL PRINTER IS READY
(1) 013222 100375 BPL STYPEC //
(1) 013224 116677 000002 165720 MOV#R 2(SP),#0TPB //LOAD CHAR TO BE TYPED INTO DATA REG.

```





```

(1) 014334 001413          SEQ      128          JIF NOT:  R0
(1) 014336 005937 001174    118:    TST      MSGTYPE  JFINISHED LAST MESSAGE?
(1) 014342 001375          BNE      118          JIF NOT:  WAIT
(1) 014344 017637 000000 001176    MOV      #4(SP),SFATAL JGET ERROR #
(1) 014352 062766 000002 000004    ADD      #2,4(SP)      JIBUMP RETURN ADDR.
(1) 014360 005237 001174          INC      MSGTYPE      JITELL APT TO TAKE ERROR
(1) 014364 105037 014410    123:    CLRR    SPFLG      JICLEAR FATAL FLAG
(1) 014370 105037 014407          CLRR    SLFLG      JICLEAR LOG FLAG
(1) 014374 105037 014406          CLRR    SMFLG      JICLEAR MESSAGE FLAG
(1) 014400 012601          MOV      (SP)+,R1     JIPOP STACK INTO R1
(1) 014402 012600          MOV      (SP)+,R0     JIPOP STACK INTO R0
(1) 014404 000207          RTS      PC           JIRETURN
(1) 014406 000          SMFLG:  .BYTE 0      JIMESSG, FLAG
(1) 014407 000          SLFLG:  .BYTE 0      JILOG FLAG
(1) 014410 000          SPFLG:  .BYTE 0      JIFATAL FLAG
(1) 014412          APTSIZE=200
(1) 000200          APTENV=001
(1) 000001          APTSPDOL=100
(1) 000100          APTCSUP=000
(1) 000040
6403
    
```

```

6405          .SBTTL TRAP DECODER
(1)
(2)          JI*****
(1)          JI*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)          JI*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)          JI*OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
(1)          JI*GO TO THAT ROUTINE.
(1) 014412 010046          STRAP:  MOV      R0,-(SP)      JISAVE R0
(1) 014414 016600 000002          MOV      2(SP),R0          JIGET TRAP ADDRESS
(1) 014420 005740          TST      -(R0)            JIBACKUP BY 2
(1) 014422 111000          MOV8     (R0),R0          JIGET RIGHT BYTE OF TRAP
(1) 014424 006300          ASL      R0              JIPOSITION FOR INDEXING
(1) 014426 016000 014446          MOV      STRPAD(R0),R0     JIINDEX TO TABLE
(1) 014432 000200          RTS      R0              JI*GO TO ROUTINE
(1)
(1)          JI*THIS IS USE TO HANDLE THE "GETPRI" MACRO
(1) 014434 011646          STRAP:  MOV      (SP),-(SP)  JI*MOVE THE PC DOWN
(1) 014436 016666 000004 000002          MOV      4(SP),2(SP)      JI*MOVE THE PSW DOWN
(1) 014444 000002          RTI      JI*RESTORE THE PSW
(1)
(1)          .SBTTL TRAP TABLE
(1)
(1)          JI*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(1)          JI*BY THE "TRAP" INSTRUCTION.
(1)
(1)          JI ROUTINE
(1)          JI-----
(1) 014446 014434          STPAD:  .WORD  STRAP
(1) 014450 013004          STYPE  JICALLTYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
(1) 014452 012002          STPOC  JICALLTYPOC     TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(1) 014454 012056          STVPS  JICALLTYVPS     TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
(1) 014456 012616          STVPON JICALLTYVPON     TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
(1) 014460 011300          STVPS  JICALLTYVPS     TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
(1) 014462 011224          STVPSN JICALLTYVPSN     TRAP+6(104406) TYPE BINARY (ASCII) NUMBER
(1)
(1) 014464 013336          BGTSHR JICALLBGTSHR    TRAP+7(104407) GET SOFT-SWR SETTING
(1)
(1) 014466 013266          BCKSHR JICALLBCKSHR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
(1) 014470 013550          BRDCHR JICALLBRDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
(1) 014472 013670          BRDLIN JICALLBRDLIN   TRAP+12(104412) TTY TYPEIN STRING ROUTINE
(1) 014474 014042          BRDOCT JICALLBRDOCT   TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
6406
6407          000001          .END
    
```

ABASE	# 170440	5553#	5574	5653	5654	5655	5656	6301
ACDW1	# 000000	5574						
ACDW2	# 000000	5574						
ACPUOP	# 000000	5574						
ADBR	# 007030	6186	6312#					
ADCS	# 007026	6180*	6183*	6184	6311#			
ADDDK	# 001440	5563	5608#					
ADDW0	# 000000	5574						
ADDW1	# 000000	5574						
ADDW10	# 000000	5574						
ADDW11	# 000000	5574						
ADDW12	# 000000	5574						
ADDW13	# 000000	5574						
ADDW14	# 000000	5574						
ADDW15	# 000000	5574						
ADDW2	# 000000	5574						
ADDW3	# 000000	5574						
ADDW4	# 000000	5574						
ADDW5	# 000000	5574						
ADDW6	# 000000	5574						
ADDW7	# 000000	5574						
ADDW8	# 000000	5574						
ADDW9	# 000000	5574						
ADEVCT	# 000000	5574						
ADEVM	# 000000	5574						
ADJR34	# 010376	6054	6113	6355#				
ADJR35	# 010431	6056	6356#					
ADJR36	# 010464	6058	6357#					
ADJR37	# 010517	6060	6358#					
ADJR46	# 010552	6054	6087	6359#				
ADJR47	# 010605	6056	6360#					
ADJR48	# 010640	6058	6361#					
ADJR49	# 010673	6060	6362#					
AENV	# 000000	5574						
AENVH	# 000000	5574						
AFATAL	# 000000	5574						
AMADR1	# 000000	5574						
AMADR2	# 000000	5574						
AMADR3	# 000000	5574						
AMADR4	# 000000	5574						
AMAMS1	# 000000	5574						
AMAMS2	# 000000	5574						
AMAMS3	# 000000	5574						
AMAMS4	# 000000	5574						
AMSGAD	# 000000	5574						
AMSGLG	# 000000	5574						
AMSGTY	# 000000	5574						
AMTYP1	# 000000	5574						
AMTYP2	# 000000	5574						
AMTYP3	# 000000	5574						
AMTYP4	# 000000	5574						
APAS3	# 000000	5574						
APRIOR	# 000000	5574						
APTCSJ	# 000040	6399	6402#					
APTENV	# 000001	6392	6399	6402#				
APTSIZ	# 000200	5696	6402#					

APTSPO	# 000100	6399	6402#																
ASWREG	# 000000	5574																	
ATESTN	# 000000	5574																	
AUNIT	# 000000	5574																	
AUSWR	# 000000	5574																	
AVECT1	# 000000	5574																	
AVECT2	# 000000	5574																	
BADUNT	# 007004	5697*	6070	6302#	6392*														
BEGIN	# 001450	5557	5690#	6394															
BEGINA	# 001456	5689	5691#																
BEGIN1	# 001462	5687	5692#																
BIT0	# 000001	5551#	5781	6304															
BIT00	# 000001	5551#																	
BIT01	# 000002	5551#																	
BIT02	# 000004	5551#																	
BIT03	# 000010	5551#																	
BIT04	# 000020	5551#																	
BIT05	# 000040	5551#																	
BIT06	# 000100	5551#																	
BIT07	# 000200	5551#																	
BIT08	# 000400	5551#	6391																
BIT09	# 001000	5551#	6391	6392															
BIT1	# 000002	5551#																	
BIT10	# 002000	5551#	6392																
BIT11	# 004000	5551#	5812	5821	5839	5847	5867	5876	5897	5907	6022	6391							
BIT12	# 010000	5551#																	
BIT13	# 020000	5551#	6392																
BIT14	# 040000	5551#	6391																
BIT15	# 100000	5551#	5773																
BIT2	# 000004	5551#																	
BIT3	# 000010	5551#	6021	6219															
BIT4	# 000020	5551#																	
BIT5	# 000040	5551#																	
BIT6	# 000100	5551#																	
BIT7	# 000200	5551#	6161	6165	6182														
BIT8	# 000400	5551#																	
BIT9	# 001000	5551#																	
BPTVEC	# 000014	5551#																	
CALDAC	# 006104	6054	6056	6058	6060	6130#													
CKSWR	# 104410	6273	6285	6391	6392	6405#													
COMPAR	# 006320	6060	6049	6093	6119	6142	6224#												
CONVRT	# 006302	6037	6046	6090	6116	6138	6177#												
CR	# 000015	5551#	6399																
CRLF	# 000200	5551#	6399																
CSPACE	# 006422	6080	6114	6202#															
DACBAD	# 007002	6077*	6106*	6132*	6301#	6386													
DAC0	# 001422	5653#	5710*	5705	5797*	5798	5805*	5806	5813*	5814	5821*	5822*	5911*	5916					
DAC1	# 001424	5940*	5943	5982*	6054	6252	6275*	6292*	6382										
DAC2	# 001426	56540	5715*	5718*	5786	5825*	5826	5832*	5833	5840*	5841	5847*	5848*	5912*					
DAC3	# 001430	5922	5949*	5952	5983*	6056	6254	6276*	6293*	6383									
		56550	5716*	5719*	5787	5852*	5853	5860*	5861	5868*	5869	5876*	5878*	5913*					
		5923	5959*	5962	5984*	6058	6256	6277*	6294*	6384									
		56560	5717*	5720*	5788	5882*	5883	5890*	5891	5898*	5899	5907*	5908*	5914*					
		5934	5967*	5971	5985*	6024*	6060	6258	6278*	6295*	6385								
DDISP	# 177570	5551#	5574	5696															
DF0	# 011204	5502	5508	5600	5606	5612	5619	5625	5631	5637	5643	6387#							











MAINDEC-11-DVAAA-A DVAAA,P11	AAV11 CROSS REFERENCE	DIAGNOSTIC TABLE --	MACY11 27(732) PERMANENT SYMBOLS	12-AUG-76	10132	PAGE 90-1	SEQ 0071								
6060	6062	6067	6070	6075	6077	6078	6079	6080	6085	6089	6092	6104	6105		
6106	6107	6108	6111	6115	6118	6130	6131	6132	6133	6135	6136	6141	6154	6158	
6162	6168	6177	6180	6182	6193	6203	6213	6224	6225	6226	6227	6233	6234	6238	
6239	6250	6252	6254	6256	6258	6271	6274	6275	6276	6277	6278	6283	6284	6292	
6293	6294	6295	6296	6309	6300	6391	6392	6393	6394	6398	6399	6400	6401	6402	
6405															
MOV8	6155	6157	6389	6390	6391	6392	6398	6399	6400	6401	6402	6405			
NEG	6390	6390													
NOP	6062														
RESET	5494	5942	5951	5961	5970	6062									
ROL	6389	6398	6401												
ROR	6190														
RTI	5696	6389	6390	6391	6392	6394	6398	6399	6400	6401	6405				
RT8	5721	6099	6125	6148	6172	6194	6217	6236	6240	6266	6299	6393	6399	6402	6405
SEC	6399														
SUB	5822	5848	5878	5908	6145	6146	6228	6390	6392	6402					
SWAB	6178														
TRAP	6405														
YST	5707	5730	5732	5734	5747	5750	5756	5760	5762	5785	5786	5787	5788	5822	5848
	5878	5908	6016	6054	6056	6058	6060	6064	6264	6390	6391	6392	6393	6398	6399
	6400	6401	6402	6405											
T8T8	6184	6205	6390	6391	6399	6400	6402								
.ABCI1	5574	6341	6342	6375	6378										
.ABCIZ	5574	6062	6327	6328	6329	6330	6331	6332	6333	6334	6335	6336	6337	6338	6339
	6340	6343	6344	6345	6347	6348	6349	6350	6351	6355	6356	6357	6358	6359	6360
	6361	6362	6367	6368	6370	6371	6393	6393	6400						
.BLKB	6400														
.BLKW	6390														
.BYTE	5574	5768	5769	6062	6346	6374	6376	6377	6379	6389	6392	6398	6400	6402	
.DSABL	6400														
.ENABL	4	5541	6400												
.END	6407														
.ENDC	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776	5783
	5789	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838	5843
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893	5896
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963	5966
	5972	5977	5980	6015	6017	6020	6028	6033	6041	6044	6050	6054	6056	6058	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6400	6401
	6402	6405													
.EQUIV	5551														
.EVEN	5574	6380	6393	6396	6402										
.IF	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776	5783
	5789	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838	5843
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893	5896
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963	5966
	5972	5977	5980	6015	6017	6020	6028	6033	6041	6044	6050	6054	6056	6058	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6400	6401
	6402	6405													
.IFF	5551	5556	5571	5573	5574	5576	5579	5579	5774	5774	5783	5789	5795	5800	5803
	5808	5811	5814	5821	5823	5828	5830	5835	5838	5843	5847	5848	5850	5855	5858
	5858	5863	5866	5871	5876	5880	5885	5888	5893	5896	5901	5907	5908	5910	5918
	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963	5966	5972	5977	5980	6015
	6017	6020	6028	6033	6041	6044	6050	6054	6058	6060	6062	6094	6120	6143	6147
	6147	6389	6390	6391	6392	6393	6394	6398	6399	6400	6401	6402	6405		
.IFT	6391	6392	6400	6401											
.IFT7	6391	6392	6400	6401											

MAINDEC-11-DVAAA-A DVAAA,P11	AAV11 CROSS REFERENCE	DIAGNOSTIC TABLE --	MACY11 27(732) PERMANENT SYMBOLS	12-AUG-76	10132	PAGE 90-2	SEQ 0072									
.IIF	5550	5556	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776	5783			
.IRP	5576	5695	5783	5795	5800	5811	5821	5822	5823	5830	5838	5847	5848	5850	5858	
	5866	5876	5878	5880	5888	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	
	6020	6033	6044	6054	6056	6058	6060	6390	6391	6392	6394	6401	6402			
.LIST	2	5537	5539	5549	5551	5556	5557	5574	5576	5695	5696	5727	5783	5795	5803	
	5811	5821	5822	5823	5830	5838	5847	5868	5850	5858	5866	5876	5878	5880	5888	
	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054	6056	
	6058	6060	6062	6247	6321	6391	6392	6400	6405							
.MACRO	43	68	169	312	491	898	937	1009	1173	1239	1271	1295	1499	1440	1471	
	1519	1531	1578	1614	1647	1660	1681	1694	1718	1772	1839	1865	1922	1969	2002	
	2032	2088	2096	2155	2389	2635	2830	2917	3040	3136	3213	3320	3715	3806	3861	
	3983	4138	4210	4263	4385	4423	4486	4583	4667	4706	4768	4806	4850	4953	4997	
	5253	5299	5344	5420	5556	5574	5678	5672	5696	5990	6352	6364	6405			
	5544	5545	5546	5547	5548	5551	5574	5696								
.MCALL	5574															
.MEXIT																
.NLIST	1	3	5530	5540	5551	5556	5557	5574	5576	5695	5696	5722	5783	5795	5803	
	5811	5821	5822	5823	5830	5838	5847	5868	5850	5858	5866	5876	5878	5880	5888	
	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054	6056	
	6058	6060	6062	6242	6317	6391	6392	6400	6405							
.PAGE	5574															
.REPT	5557															
.SBYTL	5551	5556	5571	5573	5574	5576	5695	5696	5723	5724	5725	5726	5741	5783	5795	5803
	5811	5821	5822	5823	5830	5838	5847	5868	5850	5858	5866	5876	5878	5880	5888	5896
	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054	6056	6060
	6058	6060	6062	6073	6102	6120	6152	6175	6200	6222	6243	6244	6245	6246	6248	6248
	6249	6281	6318	6319	6320	6325	6399	6390	6391	6392	6393	6394	6398	6399	6400	6400
	6401	6402	6405													
.TITLE	5550															
.WORD	5557	5571	5573	5574	6062	6393	6394	6398	6399	6401	6402	6405				

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0  
 \*DVAAA, DVAAA/CRF=DVAAA, SML, DVAAA, P11  
 RUN-TIME: 41 30 5 SECONDS  
 RUN-TIME RATIO: 820/97=8.4  
 CORE USED: 34K (67 PAGES)