

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 DDT 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
5858 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 SHOOD
(1)  %
(1)  %THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SVS%AC
(1)  %PACKAGE (MAINDEC-11-DZBAC-C1), MAR 24, 1976.
(1)  %
5551 .SBTTL  BASIC DEFINITIONS
(1)
(1)  %INITIAL ADDRESS OF THE STACK POINTER *** 11PP ***
(1)  STACK= 1100
(1)  .EQUIV  EMY,ERROR      %%BASIC DEFINITION OF ERROR CALL
(1)  .EQUIV  IOY,SCOPE     %%BASIC DEFINITION OF SCOPE CALL
(1)
(1)  %%MISCELLANEOUS DEFINITIONS
(1)  MT= 11                %%CODE FOR HORIZONTAL TAB
(1)  LP= 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)  STKLMY= 177774        %%STACK LIMIT REGISTER
(1)  PIRQ= 177772         %%PROGRAM INTERRUPT REQUEST REGISTER
(1)  DSWR= 177570        %%HARDWARE SWITCH REGISTER
(1)  DDISP= 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)  SW14= 40000
(1)  SW13= 20000
(1)  SW12= 10000

```

```

(1)  000000      SW11= 4000
(1)  002000      SW10= 2000
(1)  001000      SW09= 1000
(1)  000400      SW08= 400
(1)  000200      SW07= 200
(1)  000100      SW06= 100
(1)  000040      SW05= 40
(1)  000020      SW04= 20
(1)  000010      SW03= 10
(1)  000004      SW02= 4
(1)  000002      SW01= 2
(1)  000001      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  BIT08,BIT8
(1)  .EQUIV  BIT07,BIT7
(1)  .EQUIV  BIT06,BIT6
(1)  .EQUIV  BIT05,BIT5
(1)  .EQUIV  BIT04,BIT4
(1)  .EQUIV  BIT03,BIT3
(1)  .EQUIV  BIT02,BIT2
(1)  .EQUIV  BIT01,BIT1
(1)  .EQUIV  BIT00,BIT0
(1)
(1)  %%BASIC "CPU" TRAP VECTOR ADDRESSES
(1)  ERRVEC= 4           %%TIME OUT AND OTHER ERRORS
(1)  RESVEC= 10         %%RESERVED AND ILLEGAL INSTRUCTIONS
(1)  YBITVEC= 14        %%"Y" BIT
(1)  TRTVEC= 14        %%TRACE TRAP

```

```

(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          .SRCTL 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) ;*
(1) ;*          000000          ;*
(1) ;*          ;*ALL UNUSED LOCATIONS FROM 0 - 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 OISPREG: .WORD 0          ;;SOFTWARE DISPLAY REGISTER
(1) 000176 000000 SWREG: .WORD 0          ;;SOFTWARE SWITCH REGISTER
(1) ;*
(1) ;*          .SRCTL STARTING ADDRESS(ES)
5558          JMP @*REGIN ;;JUMP TO STARTING ADDRESS OF PROGRAM
5559          JMP FULLRMP ;;JUMP TO FULL RAMP LOOP
5560          JMP STATIC ;;JUMP TO STATIC DAC CALIBRATION
5561          JMP DYNCAL ;;JUMP TO DYNAMIC DAC CALIBRATION
5562          ;*230
5563          JMP ADDOK          ;;JUMP AND ENABLE EXTENDED UNITS <16,>
5564          ;*240
5565          JMP TESTER          ;;JUMP TO TESTER SA.
5566          ;*100
5567          ;*100,200,2
5568          ;*100,200,2          ;* EVENT SAFE GUARD
5569          ;*100,200,2
    
```

```

5571          .SBTTL ACT11 HOOKS
(1)
(2)          ;;*****
(1)          ;HOOKS REQUIRED BY ACT11
(1)          ;SSVPC=,           ;SAVE PC
(1)          ;=06
(1)          SENDAD           ;1)SET LOC.46 TO ADDRESS OF SENDAD IN .SEOP
(1) 000046 005542           ;=2
(1)          ;WORD 0           ;2)SET LOC.52 TO ZERO
(1) 000752 000000           ;SSVPC
(1)          ;=1000           ;RESTORE PC
(1)          001000
5572          .SBTTL APT PARAMETER BLOCK
5573          ;;*****
(1)          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
(2)          ;;*****
(1)          ;.SX=           ;SAVE CURRENT LOCATION
(1)          ;=24           ;SET POWER FAIL TO POINT TO START OF PROGRAM
(1) 000024 000200           ;200 ;FOR APT START UP
(1)          ;=44           ;POINT TO APT INDIRECT ADDRESS PNTR.
(1) 000044 001000           ;SAPTHDR ;POINT TO APT HEADER BLOCK
(1) 000244 001000           ;.SX ;RESET LOCATION COUNTER
(2)          ;;*****
(1)          ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
(1)          ;INTERFACE SPEC.
(1)
(1)          ;SAPTHDR:
(1) 001000 000000           ;SHRTR: .WORD 0           ;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
(1) 001002 001174           ;SMBADR: .WORD SMBAL           ;ADDRESS OF APT MAILBOX (BITS 0-15)
(1) 001004 000030           ;STSTMI: .WORD 30          ;RUN TIM OF LONGEST TEST
(1) 001006 000010           ;SPASTM: .WORD 10          ;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
(1) 001010 000030           ;SUNITM: .WORD 30          ;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
(1) 001012 000031           ;.WORD SETEND-SMAIL/2 ;LENGTH MAILBOX-ETABLE(WORDS)
    
```

```

5574          .SBTTL COMMON TAGS
(1)          ;;*****
(2)          ;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
(1)          ;USED IN THE PROGRAM.
(1)          ;=1100
(1)          SCMTAG:           ;START OF COMMON TAGS
(1) 001100 000000           ;.WORD 0
(1) 001102 000000           ;STSTMI: .BYTE 0           ;CONTAINS THE TEST NUMBER
(1) 001103 000000           ;SERFLG: .BYTE 0           ;CONTAINS ERROR FLAG
(1) 001104 000000           ;SICNT: .WORD 0           ;CONTAINS SUBTEST ITERATION COUNT
(1) 001106 000000           ;SLPADR: .WORD 0           ;CONTAINS SCOPE LOOP ADDRESS
(1) 001110 000000           ;SLPERR: .WORD 0           ;CONTAINS SCOPE RETURN FOR ERRORS
(1) 001112 000000           ;SERTTL: .WORD 0           ;CONTAINS TOTAL ERRORS DETECTED
(1) 001114 000000           ;STEMM: .BYTE 0           ;CONTAINS ITEM CONTROL BYTE
(1) 001115 001           ;SERMAX: .BYTE 1           ;CONTAINS MAX. ERRORS PER TEST
(1) 001116 000000           ;SERRPC: .WORD 0           ;CONTAINS PC OF LAST ERROR INSTRUCTION
(1) 001120 000000           ;SGADR: .WORD 0           ;CONTAINS ADDRESS OF 'GOOD' DATA
(1) 001122 000000           ;SBADR: .WORD 0           ;CONTAINS ADDRESS OF 'BAD' DATA
(1) 001124 000000           ;SGDAT: .WORD 0           ;CONTAINS 'GOOD' DATA
(1) 001126 000000           ;SBDAT: .WORD 0           ;CONTAINS 'BAD' DATA
(1) 001130 000000           ;.WORD 0
(1) 001132 000000           ;.WORD 0
(1) 001134 000           ;SAUTOB: .BYTE 0           ;AUTOMATIC MODE INDICATOR
(1) 001135 000           ;SINTAG: .BYTE 0           ;INTERRUPT MODE INDICATOR
(1) 001136 000000           ;.WORD 0
(1) 001140 177570           ;SWR: .WORD DSWR           ;ADDRESS OF SWITCH REGISTER
(1) 001142 177570           ;DISPLAY: .WORD DDISP          ;ADDRESS OF DISPLAY REGISTER
(1) 001144 177560           ;STKS: 177560             ;TTY KBD STATUS
(1) 001146 177562           ;TKR: 177562             ;TTY KBD BUFFER
(1) 001150 177564           ;STPS: 177564           ;TTY PRINTER STATUS REG. ADDRESS
(1) 001152 177566           ;STPB: 177566           ;TTY PRINTER BUFFER REG. ADDRESS
(1) 001154 000           ;SNULL: .BYTE 0           ;CONTAINS NULL CHARACTER FOR FILLS
(1) 001155 002           ;SFILLS: .BYTE 2           ;CONTAINS # OF FILLER CHARACTERS REQUIRED
(1) 001156 012           ;SFILLC: .BYTE 12          ;INSERT FILL CHARS. AFTER A "LINE FEED"
(1) 001157 000           ;STPFLG: .BYTE 0           ;"TERMINAL AVAILABLE" FLAG (BIT-0?=>0=YES)
(1) 001160 000000           ;TIMES: 0                 ;MAX. NUMBER OF ITERATIONS
(1) 001162 000000           ;SESCAPE: 0               ;ESCAPE ON ERROR ADDRESS
(1) 001164 177607 000377           ;SBELL: .ASCIIZ <207><377><377> ;CODE FOR BELL
(1) 001170 077           ;SQUES: .ASCII /?/         ;QUESTION MARK
(1) 001171 015           ;SCRLF: .ASCII <15>        ;CARRIAGE RETURN
(1) 001172 000012           ;SLF: .ASCIIZ <12>        ;LINE FEED
(2)          ;;*****
(2)          .SBTTL APT MAILBOX-ETABLE
(2)          ;;*****
(2)          ;EVEN
(2) 001174           ;SMAIL:           ;APT MAILBOX
(2) 001174 000000           ;SMSGTY: .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           ;SDEVCT: .WORD ADEVCT       ;DEVICE COUNT
(2) 001206 000000           ;SUNIT: .WORD AUNIT         ;I/O UNIT NUMBER
(2) 001210 000000           ;SMGADR: .WORD MSGADR       ;MESSAGE ADDRESS
    
```

```

(2) 001212 000000 $MSGLG1 .WORD AMAGLG ;;MESSAGE LENGTH
(2) 001214 000000 $ETABLE:
(2) 001214 000 $ENV1 .BYTE AENV ;;APT ENVIRONMENT TABLE
(2) 001215 000 $ENVM1 .BYTE AENVH ;;ENVIRONMENT BYTE
(2) 001216 000000 $SWREG1 .WORD ASWREG ;;ENVIRONMENT MODE BITS
(2) 001220 000000 $USWR1 .WORD AUSWR ;;APT SWITCH REGISTER
(2) 001222 000000 $CPUOP1 .WORD ACPUPOP ;;USER SWITCHES
(2) ;;CPU TYPE, OPTIONS
(2) ;; RITS 15=11=CPU TYPE
(2) ;; 11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(2) ;; 11/70=06,PDQ=07,G=10
(2) ;;
(2) ;; BIT 10=REAL TIME CLOCK
(2) ;; BIT 9=FLOATING POINT PROCESSOR
(2) ;; BIT 8=MEMORY MANAGEMENT
(2) 001224 000 $MAMS11 .BYTE AMAMS1 ;;HIGH ADDRESS,M.S. BYTE
(2) 001225 000 $MTYP11 .BYTE AMTYP1 ;;MEM. TYPE,BLK#1
(2) ;; MEM. TYPE BYTE -- (HIGH BYTE)
(2) ;; 900 NSEC CORE=001
(2) ;; 300 NSEC RTPOLAR=002
(2) ;; 500 NSEC MOS=003
(2) 001226 000000 $MADR11 .WORD AMADR1 ;;HIGH ADDRESS,BLK#1
(2) ;; MEM.LAST ADDR,#3 BYTES,THIS WORD AND LOW OF "TYPE" ABOVE
(2) 001230 000 $MAMS21 .BYTE AMAMS2 ;;HIGH ADDRESS,M.S. BYTE
(2) 001231 000 $MTYP21 .BYTE AMTYP2 ;;MEM. TYPE,BLK#2
(2) 001232 000000 $MADR21 .WORD AMADR2 ;;MEM.LAST ADDRESS,BLK#2
(2) 001234 000 $MAMS31 .BYTE AMAMS3 ;;HIGH ADDRESS,M.S. BYTE
(2) 001235 000 $MTYP31 .BYTE AMTYP3 ;;MEM. TYPE,BLK#3
(2) 001236 000000 $MADR31 .WORD AMADR3 ;;MEM.LAST ADDRESS,BLK#3
(2) 001240 000 $MAMS41 .BYTE AMAMS4 ;;HIGH ADDRESS,M.S. BYTE
(2) 001241 000 $MTYP41 .BYTE AMTYP4 ;;MEM. TYPE,BLK#4
(2) 001242 000000 $MADR41 .WORD AMADR4 ;;MEM.LAST ADDRESS,BLK#4
(2) 001244 000000 $VECT11 .WORD AVECT1 ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
(2) 001246 000000 $VECT21 .WORD AVECT2 ;;INTERRUPT VECTOR#2BUS PRIORITY#2
(2) 001250 170440 $BASE1 .WORD ABASE ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
(2) 001252 000000 $DEVH1 .WORD ADEVH ;;DEVICE MAP
(2) 001254 000000 $CTD11 .WORD ACD11 ;;CONTROLLER DESCRIPTION WORD#1
(2) 001256 .MEXIT
    
```

```

(1) .S0TTL ERROR POINTER TABLE
(1)
(1) ;;THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
(1) ;;THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
(1) ;;LOCATION ITEMS. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
(1) ;;NOTE1: IF ITEMS IS 0 THE ONLY PERTINENT DATA IS (SERRPC).
(1) ;;NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
(1) ;;
(1) ;; EM ;;POINTS TO THE ERROR MESSAGE
(1) ;; DH ;;POINTS TO THE DATA HEADER
(1) ;; DT ;;POINTS TO THE DATA
(1) ;; DF ;;POINTS TO THE DATA FORMAT
(1)
(1) 001256 $ERRTB1
5575
5577
5578
5579 001256 007122 ITEM 1
5580 001260 010322 EM1 ;;BUS TIME-OUT WHEN REF. A DAC ADDRESS
5581 001262 011112 DH2 ;;ERRPC BUSADR
5582 001264 011204 DT1 ;;ERRPC $DDAT
5583 DF0
5584
5585 001266 007176 ITEM 2
5586 001270 010267 EM2 ;;DAC #0 REGISTER IN ERROR
5587 001272 011120 DH1 ;;ERRPC BUSADR GOOD BAD
5588 001274 011204 DT2 ;;ERRPC DAC0 $DDAT $BDDAT
5589 DF0
5590
5591 001276 007225 ITEM 3
5592 001300 010267 EM3 ;;DAC #1 REGISTER IN ERROR
5593 001302 011132 DH1 ;;ERRPC BUSADR GOOD BAD
5594 001304 011204 DT3 ;;ERRPC DAC1 $DDAT $BDDAT
5595 DF0
5596
5597 001306 007254 ITEM 4
5598 001310 010267 EM4 ;;DAC #2 REGISTER IN ERROR
5599 001312 011144 DH1 ;;ERRPC BUSADR GOOD BAD
5600 001314 011204 DT4 ;;ERRPC DAC2 $DDAT $BDDAT
5601 DF0
5602
5603 001316 007303 ITEM 5
5604 001320 010267 EM5 ;;DAC #3 REGISTER IN ERROR
5605 001322 011156 DH1 ;;ERRPC BUSADR GOOD BAD
5606 001324 011204 DT5 ;;ERRPC DAC3 $DDAT $BDDAT
5607 DF0
5608
5609 001326 007332 ITEM 6
5610 001330 010337 EM6 ;;SELECTED DAC OFFSET POT IS NOT ADJUSTED CORRECTLY
5611 001332 011170 DH6 ;;ERRPC BUSADR EXPECT WAS SPREAD
5612 001334 011204 DT6 ;;ERRPC DACBAD $DDAT $BDDAT SPREAD
5613 DF0
    
```

ADDRESS	DATA1	DATA2	DATA3	DESCRIPTION
5615				ITEM 7
5616	001336	007413		EM7
5617	001340	010337		DM6
5618	001342	011170		DT6
5619	001344	011204		DF0
5620				
5621				ITEM 10
5622	001346	007472		EM10
5623	001350	010337		DM6
5624	001352	011170		DT6
5625	001354	011204		DF0
5626				
5627				ITEM 11
5628	001356	007537		EM11
5629	001360	010337		DM6
5630	001362	011170		DT6
5631	001364	011204		DF0
5632				
5633				ITEM 12
5634	001366	007574		EM12
5635	001370	010337		DM6
5636	001372	011170		DT6
5637	001374	011204		DF0
5638				
5639				ITEM 13
5640	001376	007631		EM13
5641	001400	010267		DM1
5642	001402	011156		DT5
5643	001404	011204		DF0
5644				
5645				ITEM 14
5646	001406	007675		EM14
5647	001410	000000		0
5648	001412	000000		0
5649	001414	000000		0
5650				
5651	001416	000010		VADDR: 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

ADDRESS	DATA1	DATA2	DATA3	DATA4	DESCRIPTION
5606	001432	005237	007020		TESTER: INC WFTST //INDICATE TESTER MODE
5607	001436	000411			RR BEGIN: //
5608	001440	012737	000021	007006	ADDOK: MOV #17,NUMBOK //LOAD 16 MAX UNITS
5609	001446	000403			RR BEGIN: //
5610	001450	012737	000005	007006	MOV #5,NUMBOK //LOAD 4 MAX UNITS
5611	001456	005037	007020		BEGIN: CLR WFTST
5612	001462	005037	007012		BEGIN: CLR TEMP
5613	001466	005037	001420		CLR EVER
5614	001472	000005			RESET
5615					.SBTTL INITIALIZE THE COMMON TAGS
(1)					//CLEAR THE COMMON TAGS (SCMTAG) AREA
(1)	001474	012706	001100		MOV #SCMTAG,R6 //FIRST LOCATION TO BE CLEARED
(1)	001500	005026			CLR (R6)+ //CLEAR MEMORY LOCATION
(1)	001502	022706	001140		CMP #SWR,R6 //DONE?
(1)	001504	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	000020	MOV #SSCOPE,#IOTVEC //IOT VECTOR FOR SCOPE ROUTINE
(1)	001522	012737	000340	000022	MOV #340,#IOTVEC+2 //LEVEL 7
(1)	001530	012737	012006	000030	MOV #SEERROR,#EMTVEC //EMT VECTOR FOR ERROR ROUTINE
(1)	001536	012737	000340	000032	MOV #340,#EMTVEC+2 //LEVEL 7
(1)	001544	012737	014412	000034	MOV #STRAP,#TRAPVEC //TRAP VECTOR FOR TRAP CALLS
(1)	001552	012737	000340	000036	MOV #340,#TRAPVEC+2 //LEVEL 7
(1)	001560	012737	012536	000024	MOV #SPWRNN,#PWRVEC //POWER FAILURE VECTOR
(1)	001566	012737	000340	000026	MOV #340,#PWRVEC+2 //LEVEL 7
(1)	001574	005037	001100		MOV #TIMES //INITIALIZE NUMBER OF ITERATIONS
(1)	001600	005037	001102		CLR ESCAPE //CLEAR THE ESCAPE ON ERROR ADDRESS
(1)	001604	112737	000001	001115	MOVR #1,SEMAX //ALLOW ONE ERROR PER TEST
(1)	001612	012737	001612	001106	MOV #.,SLPADR //INITIALIZE THE LOOP ADDRESS FOR SCOPE
(1)	001620	012737	001620	001110	MOV #.,SLPERR //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	000004	MOV #668,#ERRVEC //SET UP ERROR VECTOR
(2)	001640	012737	177570	001140	MOV #DSWR,SWR //SETUP FOR A HARDWARE SWICH REGISTER
(2)	001646	012737	177570	001142	MOV #DISP,DISPLAY //FIND A HARDWARE DISPLAY REGISTER
(2)	001654	022777	177777	177256	CMP #1,PSWR //TRY TO REFERENCE HARDWARE SWR
(2)	001662	001612			BNE #68 //BRANCH IF NO TIMEOUT TRAP OCCURRED
(2)					//FIND THE HARDWARE SWR IS NOT = -1
(2)	001664	000403			BR #68 //BRANCH IF NO TIMEOUT
(2)	001666	012716	001674	648:	MOV #658,(SP) //SET UP FOR TRAP RETURN
(2)	001672	000002			RTI
(2)	001674	012737	000176	001140	MOV #SWREG,SWR //POINT TO SOFTWARE SWR
(2)	001702	012737	000174	001142	MOV #DISPREG,DISPLAY //RESTORE ERROR VECTOR
(2)	001710	012637	000004	668:	MOV (SP),#ERRVEC
(2)					
(2)	001714	005037	001202		CLR SPASS //CLEAR PASS COUNT
(2)	001720	132737	000200	001215	BITR #APTSIZE,SENVH //TEST USER SIZE UNDER APT
(2)	001726	001403			BEQ #67 //YES,USE NON-APT SWITCH
(2)	001730	012737	001216	001140	MOV #SWREG,SWR //NO,USE APT SWITCH REGISTER
(2)	001736				
5697	001736	005037	007004		CLR BADUNT //RESET BAD INDICATOR
5698	001742	000137	002044		JMP INTR

```

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 501 MOV R2,(R1)+ ;LOAD .+2
5705 001760 005021 CLR (R1)+ ;LOAD HALT
5706 001762 010102 MOV R1,R2 ;LOAD R2
5707 001764 005722 TST (R2)+ ;BUMP R2
5708 001766 020227 001002 CMP R2,#1002 ;TEST FOR LAST
5709 001772 001371 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 RTS PC ;EXIT
5722
5729 002044 004737 001746 INIT: JSR PC,LDTRAP
5730 002050 005737 007012 TST TEMP ;TEST IF START OR RESTART
5731 002054 001012 RNF MFTST ;RESTART
5732 002056 005737 000042 TST #042 ;TEST IF MONITOR
5733 002062 001007 BNE MFTST ;BR IF NOT
5734 002064 005737 007020 TST WFTST ;TEST IF ON TESTER
5735 002070 001402 BEQ 19 ;BR IF NOT
5736 002072 104401 TYPE MSGSW ;TELL OPERATOR ABOUT TESTER SWITCHES
5737 002074 010011 19: TYPE ;CALL MESSAGE PRINTER VIA 'EMT'
5738 002076 104401 TYPE ;TYPE PROGRAM HEADER.
5739 002100 007040 TITLE
    
```

```

5741                                     ;SRTTL DETERMINE THE NUMBER OF AAV11 ON THIS SYSTEM
5742
5743 002102 013737 001250 001126 MTEST: MOV $BASE,$DDAT ;GET THE BASE ADDRESS
5744 002110 005037 007010 CLR MASKNM ;CLEAR UNIT #
5745 002114 005037 001206 CLR $UNIT ;LOAD TRAP RETURN
5746 002120 012737 002164 000004 19: MOV #2$,ERRVEC ;TEST IF ADDR EXISTS
5747 002126 005777 176774 TST $RRODAT ;UPDATE THE BUS ADDRESS
5748 002132 063737 001416 001126 ADD VADDR,$DDAT ;UPDATE UNIT COUNT
5749 002140 005237 001206 INC $UNIT ;TEST IF "NO NOT SIZE"
5750 002144 005737 001214 TST $ENV ;BR IF NO SIZING
5751 002150 100413 BMI 38 ;TEST IF MAX. NUMBER
5752 002152 023737 007006 001206 CMP NUMROK,$UNIT ;BR IF NOT
5753 002160 001362 BR 38 ;BR IF MAX.
5754 002162 000406 BR 38 ;BR IF MAX.
5755 002164 022626 29: CMP ($P)+,($P)+ ;CLEAN THE STACK
5756 002166 005737 001206 TST $UNIT ;TEST IF ANY EXIST
5757 002172 001002 RNF 38 ;BR IF SOME ARE THERE
5758 002174 104001 ERROR 1 ;BASE ADDRESS CAUSED AN BUS TRAP
5759 002176 000403 BR TST1 ;IS $BASE CORRECT??
5760 002200 005737 001420 30: TST EVFR ;TEST IF # HAS BEEN REPORTED
5761 002204 100422 BMI 48 ;BR IF IT WAS
5762 002206 005737 007020 TST WFTST ;TEST IF TESTER NONE
5763 002212 001010 BNE 68 ;BR IF TESTER
5764 002214 104401 TYPE ;TELL OPERATOR THE # OF AAV11'S
5765 002216 010224 FOUND1 ;
5766 002220 013746 001206 MOV $UNIT,-($P)
5767 002224 104403 TYPOS ;
5768 002226 002 .BYTE 2
5769 002227 000 .BYTE 0
5770 002230 104401 TYPE FOUND2 ;
5771 002232 010250 MOV $UNIT,EVER ;SAVE THE # OF AAV11'S FOR LATER
5772 002234 013737 001206 001420 69: BIS #RT15,EVER ;SET "REPORTED # FLAG"
5773 002242 052737 100000 001420 BR 58 ;
5774 002250 000405 58: CMPB EVER,$UNIT ;TEST IF ANY HAVE GONE AWAY
5775 002252 123737 001420 001206 48: BEQ 58 ;BR IF ALL ARE STILL HERE
5776 002260 001401 ERROR 1 ;EXISTING UNIT FAILED TO RESPOND NOW
5777 002262 104013 BR 58 ;RESET UNIT POINTER
5778 002264 005037 001206 59: CLR $OEVCY ;MAKE APT HAPPY
5779 002270 005037 001204 CLR PC,LDTRAP ;LOAD TRAP CATCHER AND BUS ADDRESSES
5780 002274 004737 001746 JSR PC,LDTRAP ;LOAD MASK NUMBER IF ERROR
5781 002300 012737 000001 007010 MOV #RT10,MASKNM
    
```



```

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
(1) 002460 012737 000100 001160
5812 002466 012737 004000 001124
5813 002474 013777 001124 176720
5814 002502 017737 176714 001126
5815 002510 023737 001124 001126
5816 002516 001401
5817 002520 104002
5818 002522 006237 001124
5819 002526 001362

//*****
//TEST 1 TEST THAT THE AAV11 RESPONDS TO THE CPU
//*****
TST1: SCOPE
MOV #10,ERRVEC //LOAD BUS TRAP RETURN
TST #DAC0 //TEST DAC #0
TST #DAC1 //TEST DAC #1
TST #DAC2 //TEST DAC #2
TST #DAC3 //TEST DAC #3
BR 20 //BR AND RSTORE LOC. 4
10: CMP (SP)+,(SP)+ //CLEAN THE STACK
ERROR 1 //ERROR, BUS TIMEOUT WHEN ADDRESSING THE AAV11
MOV #6,ERRVEC //LOAD LOC 4
JMP REMAIN //TEST IF ANY OTHER'S
20: MOV #6,ERRVEC //LOAD RETURN

//*****
//TEST 2 TEST THAT DAC0 REGISTER CAN BE CLEARED
//*****
TST2: SCOPE
CLR $GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC0 //LOAD REG
MOV #DAC0,$SDDAT //READ REG
CMP $GDDAT,$SDDAT //COMPARE
BEQ TST3 //BR IF EQUAL
ERROR 2 //ERROR, DAC0 REGISTER NOT = 0

//*****
//TEST 3 TEST THAT DAC0 REGISTER CAN BE LOADED WITH #7777
//*****
TST3: SCOPE
MOV #7777,$GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC0 //LOAD REG
MOV #DAC0,$SDDAT //READ REG
CMP $GDDAT,$SDDAT //COMPARE
BEQ TST4 //BR IF EQUAL
ERROR 2 //ERROR, DAC0 REGISTER NOT = 7777

//*****
//TEST 4 TEST THAT DAC0 REGISTER CAN HOLD A FLOATING 1 PATTERN
//*****
TST4: SCOPE
MOV #100,$TIMES //100 100 ITERATIONS
MOV #RIT11,$GDDAT //LOAD EXPECTED
10: MOV $GDDAT,#DAC0 //LOAD DAC0 REGISTER
MOV #DAC0,$SDDAT //READ THE REGISTER
CMP $GDDAT,$SDDAT //COMPARE THE DATA
BEQ 20 //BR IF SAME
ERROR 2 //ERROR, DAC0 REGISTER FAILED TO HOLD A FLOATING
20: ASR $GDDAT //CHANGE THE DATA
RNE 10 //BR AND TEST MORE DATA

//*****

```

```

5821
(4)
(4)
(3) 002530 000004
(2) 002532 012737 000100 001160
(1) 002540 012737 004000 001124
(1) 002546 013777 001124 176646
(1) 002554 017737 176642 001126
(1) 002562 023737 001124 001126
(2) 002570 001401
(1) 002572 017737 176624 001126
(1) 002600 104002
(1) 002602 013777 001124 176612
(1) 002610 006237 176606
(1) 002614 006237 001124
(1) 002620 001355
5822
(4)
(4)
(3) 002622 000004
(2) 002624 012737 000010 001160
(1) 002632 012737 007777 001124
(1) 002640 013777 001124 176554
(1) 002646 162737 000001 001124
(1) 002654 162777 000001 176540
(1) 002662 017737 176534 001126
(1) 002670 023737 001124 001126
(2) 002676 001401
(1) 002700 104002
(1) 002702 013777 001124 176512
(1) 002710 005737 001124
(2) 002714 001354
5823
(3)
(3)
(2) 002716 000004
5824 002720 005037 001124
5825 002724 013777 001124 176472
5826 002732 017737 176466 001126
5827 002740 023737 001124 001126
5828 002746 001401
5829 002750 104003
5830
(3)
(3)
(2) 002752 000004
5831 002754 012737 007777 001124
5832 002762 013777 001124 176434
5833 002770 017737 176430 001126
5834 002776 023737 001124 001126
5835 003004 001401
5836 003006 104003

//*****
//TEST 5 TEST THAT DAC0 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
//*****
TST5: SCOPE
MOV #100,$TIMES //100 100 ITERATIONS
MOV #RIT11,$GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC0 //LOAD DAC0
MOV #DAC0,$SDDAT //READ THE REGISTER
CMP $GDDAT,$SDDAT //COMPARE THE GOOD TO DAC0
BEQ 20 //BR IF THE SAME
MOV #DAC0,$SDDAT //SAVE FOR TYPEOUT
ERROR 2 //DAC0 FAILED TO HOLD A FLOATING 1 PATTERN
MOV $GDDAT,#DAC0 //LOAD DAC0 AGAIN
20: ASR #DAC0 //CHANGE THE DATA
ASR $GDDAT //CHANGE THE EXPECTED
RNE 10 //BR IF MORE DATA

//*****
//TEST 6 TEST THE "SUB" INSTRUCTION WORKS ON DAC0
//*****
TST6: SCOPE
MOV #10,$TIMES //100 10 ITERATIONS
MOV #7777,$GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC0 //LOAD DAC0
10: SUB #1,$GDDAT //SUB A VALUE
SUB #1,#DAC0 //FROM EXPECTED AND DAC0
MOV #DAC0,$SDDAT //READ THE REGISTER
CMP $GDDAT,$SDDAT //COMPARE
BEQ 20 //BR IF SAME
ERROR 2 //THE SUB INSTRUCTION FAILED ON DAC0
MOV $GDDAT,#DAC0 //LOAD THE REGISTER AGAIN
TST $GDDAT //TEST FOR MORE DATA
20: RNE 10 //BR IF MORE DATA

//*****
//TEST 7 TEST THAT DAC1 REGISTER CAN BE CLEARED
//*****
TST7: SCOPE
CLR $GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC1 //LOAD DAC1
MOV #DAC1,$SDDAT //READ REG
CMP $GDDAT,$SDDAT //COMPARE
BEQ TST10 //BR IF EQUAL
ERROR 3 //ERROR, DAC1 REGISTER NOT = 0

//*****
//TEST 10 TEST THAT DAC #1 REGISTER CAN BE LOADED WITH #7777
//*****
TST10: SCOPE
MOV #7777,$GDDAT //LOAD EXPECTED
MOV $GDDAT,#DAC1 //LOAD DAC #1
MOV #DAC1,$SDDAT //READ REG
CMP $GDDAT,$SDDAT //COMPARE
BEQ TST11 //BR IF EQUAL
ERROR 3 //ERROR, 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
5847
(4)
(4)
(3) 003062 000004
(2) 003064 012737 000100 001160
(1) 003072 012737 004000 001124
(1) 003100 013777 001124 176316
(1) 003106 017737 176312 001126
(1) 003114 023737 001124 001126
(2) 003122 001407
(1) 003124 017737 176274 001126
(1) 003132 104003
(1) 003134 013777 001124 176262
(1) 003142 006277 176256
(1) 003146 006237 001124
(1) 003152 001355
5848
(4)
(4)
(3) 003154 000004
(2) 003156 012737 000010 001160
(1) 003164 012737 027777 001124
(1) 003172 013777 001124 176224
(1) 003200 162737 000001 001124
(1) 003206 162737 000001 176210
(1) 003214 017737 176204 001126
(1) 003222 023737 001124 001126
(2) 003230 001404
(1) 003232 104003
(1) 003234 013777 001124 176162
(1) 003242 005737 001124
(2) 003246 001354

//*****
//TEST 11 TEST THAT DAC #1 REGISTER CAN HOLD A FLOATING 1 PATTERN
//*****
TST111 SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #01111,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC1 //LOAD THE REGISTER
MOV #DAC1,SRDDAT //READ THE REGISTER
CMP SGDDAT,SRDDAT //COMPARE THE DATA
BEQ 2# //BR IF DATA IS SAME
ERROR 3 //ERROR, DAC #1 REGISTER FAILED TO HOLD A FLOATIN
ASR SGDDAT //CHANGE THE DATA
RNE 1# //R AND TEST MORE DATA

//*****
//TEST 12 TEST THAT DAC1 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
//*****
TST121 SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #01111,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC1 //LOAD DAC1
MOV #DAC1,SRDDAT //READ THE REGISTER
CMP SGDDAT,SRDDAT //COMPARE THE GOOD TO DAC1
BEQ 2# //BR IF THE SAME
MOV #DAC1,SRDDAT //SAVE FOR TYPEOUT
ERROR 3 //DAC1 FAILED TO HOLD A FLOATING 1 PATTERN
ASR SGDDAT,#DAC1 //LOAD DAC1 AGAIN
ASR #DAC1 //CHANGE THE DATA
RNE 1# //CHANGE THE EXPECTED
//R IF MORE DATA

//*****
//TEST 13 TEST THE "SUR" INSTRUCTION WORKS ON DAC1
//*****
TST131 SCOPE
MOV #10,STIMES //DO 10 ITERATIONS
MOV #7777,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC1 //LOAD DAC1
SUR #1,SGDDAT //SUR A VALUE
MOV #DAC1,SRDDAT //FROM EXPECTED AND DAC1
CMP SGDDAT,SRDDAT //READ THE REGISTER
BEQ 2# //COMPARE
ERROR 3 //BR IF SAME
MOV SGDDAT,#DAC1 //THE SUR INSTRUCTION FAILED ON DAC1
TST SGDDAT //LOAD THE REGISTER AGAIN
BNF 1# //TEST FOR MORE DATA
//R 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
5875
5876
(4)
(4)
(3) 003414 000004
(2) 003416 012737 000100 001160
(1) 003424 012737 004000 001124
(1) 003432 013777 001124 175766
(1) 003440 017737 175762 001126
(1) 003446 023737 001124 001126
(2) 003454 001407
(1) 003456 017737 175744 001126
(1) 003464 104004
(1) 003466 013777 001124 175732
(1) 003474 006277 175726
(1) 003500 006237 001124
(1) 003504 001355

//*****
//TEST 14 TEST THAT THE DAC #2 REGISTER CAN BE CLEARED
//*****
TST141 SCOPE
CLP SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC2 //LOAD REG
MOV #DAC2,SRDDAT //READ REG
CMP SGDDAT,SRDDAT //COMPARE
BEQ TST15 //BR IF EQUAL
ERROR 4 //ERROR, DAC #2 REGISTER NOT = 0

//*****
//TEST 15 TEST THAT THE DAC #2 REGISTER CAN BE LOADED WITH #7777
//*****
TST151 SCOPE
MOV #7777,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC2 //LOAD REG
MOV #DAC2,SRDDAT //READ REG
CMP SGDDAT,SRDDAT //COMPARE
BEQ TST16 //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
//*****
TST161 SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #01111,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC2 //LOAD DAC2 REGISTER
MOV #DAC2,SRDDAT //READ THE REGISTER
CMP SGDDAT,SRDDAT //COMPARE THE DATA
BEQ 2# //BR IF SAME
ERROR 4 //ERROR, DAC #2 REGISTER FAILED TO HOLD A FLOATIN
ASR SGDDAT //CHANGE THE DATA
RNE 1# //R AND TEST MORE DATA

//*****
//TEST 17 TEST THAT DAC2 CAN HOLD A FLOATING 1 PATTERN (DYNAMICLY)
//*****
TST171 SCOPE
MOV #100,STIMES //DO 100 ITERATIONS
MOV #01111,SGDDAT //LOAD EXPECTED
MOV SGDDAT,#DAC2 //LOAD DAC2
MOV #DAC2,SRDDAT //READ THE REGISTER
CMP SGDDAT,SRDDAT //COMPARE THE GOOD TO DAC2
BEQ 2# //BR IF THE SAME
MOV #DAC2,SRDDAT //SAVE FOR TYPEOUT
ERROR 4 //DAC2 FAILED TO HOLD A FLOATING 1 PATTERN
ASR SGDDAT,#DAC2 //LOAD DAC2 AGAIN
ASR #DAC2 //CHANGE THE DATA
RNE 1# //CHANGE THE EXPECTED
//R IF MORE DATA

```

```

5878
(4)
(4)
(3) 003506 000004
(1) 003510 012737 000010 001160
(2) 003516 012737 007777 001124
(1) 003524 017777 001124 175674
(1) 003532 162737 000001 001124
(1) 003540 162777 000001 175660
(1) 003546 017737 175654 001126
(1) 003554 023737 001124 001126
(2) 003562 001404
(1) 003564 104004
(1) 003566 013777 001124 175632
(1) 003574 005737 001124
(2) 003600 001354

```

```

5879
5880
(3)
(3)
(2) 003632 000004
5881 003604 005037 001124
5882 003610 013777 001124 175612
5883 003616 017737 175606 001126
5884 003624 023737 001124 001126
5885 003632 001401
5886 003634 104005
5887
5888
(3)
(3)
(2) 003636 000004
5889 003640 012737 007777 001124
5890 003646 013777 001124 175554
5891 003654 017737 175504 001126
5892 003662 023737 001124 001126
5893 003670 001401
5894 003672 104005
5895
5896
(3)
(3)
(2) 003674 000004
(1) 003676 012737 000100 001160
5897 003704 012737 004000 001124
5898 003712 013777 001124 175514
5899 003720 017737 175504 001126
5900 003726 023737 001124 001126
5901 003734 001401
5902 003736 104005
5903 003740 006237 001124
5904 003744 001362
5905

```

```

5907
(4)
(4)
(3) 003746 000004
(2) 003750 012737 000100 001160
(1) 003756 012737 004000 001124
(1) 003764 013777 001124 175436
(1) 003772 017737 175432 001126
(1) 004000 023737 001124 001126
(2) 004006 001407
(1) 004010 017737 175414 001126
(1) 004016 104005
(1) 004020 013777 001124 175402
(1) 004026 006277 175376
(1) 004032 006237 001124
(1) 004036 001355

```

```

5908
(4)
(4)
(3) 004040 000004
(2) 004042 012737 000010 001160
(1) 004050 012737 007777 001124
(1) 004056 013777 001124 175344
(1) 004064 162737 000001 001124
(1) 004072 162777 000001 175330
(1) 004100 017737 175324 001126
(1) 004106 023737 001124 001126
(2) 004114 001404
(1) 004116 104005
(1) 004120 013777 001124 175302
(1) 004126 005737 001124
(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      MOV      #1111,0DAC0      JLOAD DAC #0
5912      MOV      #2222,0DAC1      JLOAD DAC #1
5913      MOV      #4444,0DAC2      JLOAD DAC #2
5914      MOV      #7777,0DAC3      JLOAD DAC #3
5915      MOV      #1111,$GDDAT      JLOAD EXPECTED
5916      MOV      #0DAC0,$RDDAT      JREAD RFG
5917      CMP      $GDDAT,$RDDAT      JCOMPARE
5918      BEQ      1$      JBR IF EQUAL
5919      ERROR    2      JERROR, SELECTED DAC #0 IN ERROR
5920
5921      MOV      #2222,$GDDAT      JLOAD EXPECTED
5922      MOV      #0DAC1,$RDDAT      JREAD REG
5923      CMP      $GDDAT,$RDDAT      JCOMPARE
5924      BEQ      2$      JBR IF EQUAL
5925      ERROR    3      JERROR, SELECTED DAC #1 IN ERROR
5926
5927      MOV      #4444,$GDDAT      JLOAD EXPECTED
5928      MOV      #0DAC2,$RDDAT      JREAD REG
5929      CMP      $GDDAT,$RDDAT      JCOMPARE
5930      BEQ      3$      JBR IF SAME
5931      ERROR    4      JERROR, SELECTED DAC #2 IN ERROR
5932
5933      MOV      #7777,$GDDAT      JLOAD EXPECTED
5934      MOV      #0DAC3,$RDDAT      JREAD REG
5935      CMP      $GDDAT,$RDDAT      JCOMPARE
5936      BEQ      4$      JBR IF SAME
5937      ERROR    5      JERROR, SELECTED DAC #3 IN ERROR

```

```

5939      J)*****
(3)      J)TEST 27      TEST THAT RESET CLEARS DAC #0 REGISTER
(3)      J)*****
(2)      TST27:  SCOPE
(1)      MOV      #10,$TIMES      J100 10 ITERATIONS
5940      MOV      #-1,0DAC0      JLOAD EXPECTED
5941      CLR      $GDDAT      JLOAD EXPECTED
5942      RESET
5943      MOV      #0DAC0,$RDDAT      JREAD REG
5944      CMP      $GDDAT,$RDDAT      JCOMPARE
5945      BEQ      TST30      JBR IF EQUAL
5946      ERROR    2      JERROR, RESET FAILED TO CLEAR DAC #0
5947
5948      J)*****
(3)      J)TEST 30      TEST THAT RESET CLEARS DAC #1 REGISTER
(3)      J)*****
(2)      TST30:  SCOPE
(1)      MOV      #10,$TIMES      J100 10 ITERATIONS
5949      MOV      #-1,0DAC1      JLOAD EXPECTED
5950      CLR      $GDDAT      JLOAD EXPECTED
5951      RESET
5952      MOV      #0DAC1,$RDDAT      JREAD RFG
5953      CMP      $GDDAT,$RDDAT      JCOMPARE
5954      BEQ      TST31      JBR IF EQUAL
5955      ERROR    3      JERROR, 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)      MOV      #10,$TIMES      J100 10 ITERATIONS
5959      MOV      #-1,0DAC2      JLOAD THE REGISTER
5960      CLR      $GDDAT      JCLEAR EXPECTED
5961      RESET
5962      MOV      #0DAC2,$RDDAT      JREAD THE REGISTER
5963      BEQ      TST32      JBR IF CLEARED
5964      ERROR    4      JERROR, RESET FAILED TO CLEAR DAC #2
5965
5966      J)*****
(3)      J)TEST 32      TEST THAT RESET CLEARS DAC #3 REGISTER
(3)      J)*****
(2)      TST32:  SCOPE
(1)      MOV      #10,$TIMES      J100 10 ITERATIONS
5967      MOV      #-1,0DAC3      JLOAD THE REGISTER
5968      CLR      $RDDAT      JCLEAR THE EXPECTED
5969      MOV      #1,$TEMP
5970      RESET
5971      MOV      #0DAC3,$RDDAT      JREAD THE REGISTER
5972      BEQ      TST33      JBR IF CLEARED
5973      ERROR    5      JERROR, RESET FAILED TO CLEAR DAC #3
5974

```

```

5976 004530          REMAIN:
5977                *****
(3)                J*TEST 33      DETERMINE IF MORE AAV11'S REMAIN TO BE TESTED
(3)                *****
(2) 004530 000004    TST33:  SCOPE
(1) 004532 012737 000001 001160      MOV      #1,STIMES      J100 1 ITERATION
5978 004540 005237 001206 001420      INC      SUNIT          JUPDATE UNIT #
5979 004544 123737 001206 001420      CMPB    SUNIT,EVER     JTEST IF MORE
5980 004552 001424      REG      TST34          J1BR IF NOT
5981 004554 005237 001204      INC      SDEVCT        JAPT UNIT #
5982 004560 063737 001416 001422      ADD     VADDR,DAC0     JUPDATE 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         JCHANGE THE ERROR FLAG BIT
5987 004614 005037 001102      CLR     SYSTNM
5988 004620 000137 002306      JMP     TST1           JTEST THE NEXT UNIT
5989
6015                *****
(3)                J*TEST 34      DETERMINE IF RUNNING ON THE HARDWARE TESTER (IF NOT REPORT END OF PA
(3)                *****
(2) 004624 000004    TST34:  SCOPE
(1) 004626 012737 000001 001160      MOV     #1,STIMES     J100 1 ITERATION
6016 004634 005737 007020      TST    WFTST          JTEST IF ON TESTER
6017 004640 001002      BNE    TST35          J1BR TO TEST
6018 004642 000137 005454      JMP     SEQP

```

```

6020                *****
(3)                J*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     J100 1 ITERATION
6021 004656 012737 000010 007014      MOV     #R173,STEMP   JLOAD DAC PATTERN
6022 004664 012737 004000 001124      MOV     #R1711,SGDDAT JLOAD EXPECTED PATTERN
6023
6024 004672 013777 007014 174530 151  MOV     STEM,SDACS     JLOAD DAC REGISTER
6025 004700 017737 002120 001126      MOV     #DRIN,SRDDAT  JREAD THE REGISTER
6026 004706 042737 170377 001126      BIC    #170377,SRDDAT JMASK OFF OTHER BITS
6027 004714 023737 001124 001126      CMP    SGDDAT,SRDDAT  JCOMPARE
6028 004722 001401      BEQ    28             J1BR IF THE SAME
6029 004724 104013      ERROR  13            JDAC #3 DIGITAL OUTPUT BITS IN ERROR
6030
6031 004726 006237 001124      281  ASR    SGDDAT      JANJUST EXPECTED
6032 004732 006237 007014      ASR    STEM          JANJUST LOADED PATTERN
6033 004736 001355      BNE    18
6034
6035                *****
(3)                J*TEST 36      VERIFY THE AAV11 +15 SUPPLY
(3)                *****
(2) 004740 000004    TST36:  SCOPE
(1) 004742 012737 000001 001160      MOV     #1,STIMES     J100 1 ITERATION
6036 004750 013737 007032 001124      MOV     V5744,SGDDAT  JLOAD EXPECTED
6037 004756 004537 006302      JSR    R5,CONVRT      JSAMPLE THE CHANNEL
6038 004762 000012      12
6039 004764 013737 007034 007016      MOV     V144,SPREAD   JLOAD TOLERANCE
6040 004772 004737 006520      JSR    PC,COMPAR      JTEST IT
6041 004776 000401      BR     TST37          J1BR
6042 005000 104011      ERROR  11            J+15 VOLT SUPPLY IS WRONG
6043
6044                *****
(3)                J*TEST 37      VERIFY THE AAV11 -15 SUPPLY
(3)                *****
(2) 005002 000004    TST37:  SCOPE
(1) 005004 012737 000001 001160      MOV     #1,STIMES     J100 1 ITERATION
6045 005012 013737 007036 001124      MOV     V2034,SGDDAT  JLOAD EXPECTED
6046 005020 004537 006302      JSR    R5,CONVRT      JSAMPLE THE CHANNEL
6047 005024 000011      11
6048 005026 013737 007034 007016      MOV     V144,SPREAD   JLOAD TOLERANCE
6049 005034 004737 006520      JSR    PC,COMPAR      JTEST IT
6050 005040 000401      BR     TST40          J1BR
6051 005042 104012      ERROR  12            J-15 VOLT SUPPLY IS WRONG
6052

```



```

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 010660
(1) 005300 000016
(1)
(5)
(4)
(4)
(3) 005302 000004
(2) 005304 012737 000001 001160
(1) 005312 005737 001202
(3) 005316 001005
(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
RNE TST47 //BR IF NOT

JSR R5,OFFDAC //LOAD AND EXECUTE DAC OFFSET ADJ.
DAC2 //DAC ADDRESS
SELD2 //TYPEOUT ADDRESS
ADJR46 //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
RNE TST50 //BR IF NOT

JSR R5,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 R5,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 001005
(1)
(1) 005422 004537 005764
(1) 005424 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
RNE TST52 //BR IF NOT

JSR R5,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
RNE TST53 //BR IF NOT

JSR R5,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 R5,CALDAC //LOAD AND EXECUTE CALIBRATION
DAC3 //DAC ADDRESS
15 //CHANNEL # FOR RESULTS

```



```

6128                                     .SBTTL SUBROUTINE TO TEST THE D/A CALIBRATION
6129
6130 006104 012537 006210 CALDAC: MOV (R5)+,108 IGET BUS ADDRESS
6131 006110 017737 000074 006210 MOV #108,108 I
6132 006116 013737 006210 007002 MOV 108,DACBAD ILOAD BUS ADDRESS IF ERROR
6133 006124 012537 006150 MOV (R5)+,118 IGET CHANNEL #
6134
6135 006130 012777 007400 000052 MOV #7400,#108 ILOAD THE DAC
6136 006136 012737 007400 001124 MOV #7400,$GDDAT ILOAD THE EXPECTED VALUE
6137
6138 006144 004537 006302 18: JSR R5,CONVRT ISAMPLE THE CHANNEL
6139 006150 000013 118: 13
6140
6141 006152 012737 000003 007016 MOV #3,SPREAD ILOAD TOLERANCE
6142 006160 004737 006520 JSR PC,COMPAR ITEST THE RESULTS
6143 006164 000401 BR 28 IIBR
6144 006166 104010 ERROR 10 INON-LINEARITY IN DAC DETECTED
6145 006170 162777 000400 000012 28: SUR #400,#108 IADJUST THE CONTENTS
6146 006176 162737 000400 001124 SUR #400,$GDDAT IADJUST THE EXPECTED
6147 006204 001357 BNE 18 IIBR IF NOT DONE
6148 006206 000205 RTS R5 IEXIT
6149
6150 006210 000000 108: 0
6151
6152                                     .SBTTL SUBROUTINE TO LOAD A VOLTAGE INTO THE VOLTAGE SOURCE
6153
6154 006212 012500 SNDVLT: MOV (R5)+,R0 ILOAD THE POINTER
6155 006214 112001 28: MOV# (R0)+,R1 IGET SOME DATA
6156 006216 001421 REG 38 IRR IF TERM
6157 006220 110177 000576 MOV# R1,#FTLZ ILOAD THE DATA
6158 006224 012701 001000 MOV #1000,R1
6159 006230 005301 58: DEC R1 IDELAY
6160 006232 001376 BNF 58
6161 006234 052777 000200 000560 BIF #0177,#FILZ ISET BIT 7
6162 006242 012701 001000 MOV #1000,R1 ILOAD DELAY
6163 006246 005301 18: DEC R1 IDELAY
6164 006250 001376 BNF 18
6165 006252 042777 000200 000502 BIC #0177,#FILZ
6166 006260 000755 RR 28
6167
6168 006262 012701 000000 38: MOV #R,R1 ILOAD DELAY
6169 006266 152777 000177 000526 BISR #177,#FILZ IONISABLE BITS
6170 006274 005301 48: DEC R1 IDELAY
6171 006276 001376 BNF 48
6172 006300 000205 RTS R5 IEXIT
6173
  
```

```

6175                                     .SBTTL SUBROUTINE TO CONVERT CHANNEL N ON THE TESTER A/D
6176
6177 006302 012537 006414 CONVRT: MOV (R5)+,108 IGET THE CHANNEL #
6178 006306 000337 006414 SWAB 108
6179 006312 042737 170377 006414 RIC #170377,108 IMASK OUT OTHER BITS
6180 006320 013777 006414 000500 MOV 108,#ADCS ISELECT CHANNEL
6181 006326 005037 006416 CLR 118
6182 006332 012737 000200 006420 MOV #0177,128 ILOAD SHIFT COUNTER
6183 006340 105277 000462 18: INCR #ADCS ICONVERT CHANNEL
6184 006344 105777 000456 28: TSTB #ADCS IWAIT FOR DONE
6185 006350 100375 RPL 28
6186 006352 067737 000452 006416 ADD #ADRR,118 IUPDATE CONVERSION
6187 006360 006237 006420 ASR 128 IFINISHED ?
6188 006364 001365 BNF 18
6189 006366 000257 CCC
6190 006370 006037 006416 ROR 118
6191 006374 006237 006416 ASR 118
6192 006400 006237 006416 ASR 118 IJUSTIFY DATA
6193 006404 013737 006416 001126 MOV 118,$RDDAT ILOAD ACTUAL <ADJUSTED>
6194 006412 000205 RTS R5 IEXIT
6195
6196 006414 000000 108: 0
6197 006416 000000 118: 0
6198 006420 000000 128: 0
6199
6200                                     .SBTTL SUBROUTINE TO LOOP UNTIL OPERATOR TYPES AN "SPACE"
6201
6202 006422 104401 007746 CSPACE: TYPE, L0SPAC ITELL OPERATOR TO HIT SPACE BAR
6203 006426 012737 000014 006516 38: MOV #10,118 ILOAD DELAY COUNTER
6204 006434 005037 006514 CLR 108
6205 006440 105777 172500 18: TSTB #STKB IWAIT FOR OPERATOR
6206 006444 100410 BMT 28 IBR IF FLAG IS SET
6207 006446 005337 006514 DEC 108 IDELAY
6208 006452 001372 RNE 18 IBR IF NOT DONE
6209 006454 005337 006516 DEC 118 IDELAY AGAIN
6210 006460 001367 BNF 18
6211 006462 100014 ERROR 14 IWAKE UP OPERATOR
6212 006464 000760 BR 38 ILOOP
6213 006466 017737 172454 006514 28: MOV #STKB,108 IREAD THE CHARACTER
6214 006474 042737 177600 006514 BIC #177600,108 IMASK OF OTHER BITS
6215 006502 022737 000000 006514 CMP #40,108 ITEST FOR "SPACE"
6216 006510 001366 BNE 38 ILOOP
6217 006512 000207 RTS PC IEXIT
6218 006514 000000 108: 0
6219 006516 000010 118: BIT3
6220
  
```

```

6222 .SBTTL SURROUTINE TO COMPARE TWO LOCATIONS BY THE SPREAD
6223
6224 006520 010046 COMPARI MOV R0, -(SP) ISAVE R0
6225 006522 010146 MOV R1, -(SP) ISAVE R1
6226 006524 013700 001124 MOV 3G0DAT, R0 IGET EXPECTED VALUE
6227 006530 013701 001126 MOV 3R0DAT, R1 IGET THE UNKNOWN
6228 006534 160100 SUR R1, R0 ISUBTRACT
6229 006536 100001 BPL 85
6230 006540 005400 NEG R0
6231 006542 020037 007016 08: CMP R0, SPREAD ITEST IF DIFFERENCE IF > THAN SPREAD
6232 006546 003405 BLE 105
6233 006550 012601 98: MOV (SP)+, R1 IRESTORE R1
6234 006552 012600 MOV (SP)+, R0 IRESTORE R0
6235 006554 062716 000002 ADD #2, (SP) IMAKE AN ERROR EXIT
6236 006560 000207 RTS PC IEXIT
6237
6238 006562 012601 105: MOV (SP)+, R1
6239 006564 012600 MOV (SP)+, R0
6240 006566 000207 RTS PC IEXIT FOR GOOD LIMIT TEST
6241
6242 .SBTTL FULL SCALE RAMP ON EACH RAMP
6243
6244
6245 006570 012706 001100 FULLRMP: MOV #STACK, SP ILOAD POINTER
6246 006574 004737 001746 JSR PC, LDTRAP ILOAD BUS ADDRESS
6247 006600 013700 001422 18: MOV DAC0, R0 IGET BUS ADDRESS
6248 006604 004737 006642 JSR PC, 105 ILOAD THE RAMP ON DAC #1
6249 006610 013700 001424 MOV DAC1, R0 IGET BUS ADDRESS
6250 006614 004737 006642 JSR PC, 105 ILOAD THE RAMP ON DAC #1
6251 006620 013700 001426 MOV DAC2, R0 IGET BUS ADDRESS
6252 006624 004737 006642 JSR PC, 105 ILOAD THE RAMP ON DAC #2
6253 006630 013700 001430 MOV DAC3, R0 IGET THE BUS ADDRESS
6254 006634 004737 006642 JSR PC, 105 ILOAD THE RAMP ON DAC #3
6255 006640 000757 BR 15 IBR BACK
6256
6257 006642 005010 105: CLR (R0) ICLEAR DAC
6258 006644 062710 000010 115: ADD #10, (R0) IUPDATE THE DATA
6259 006650 005710 TST (R0) ITEST IF DONE
6260 006652 001374 RNE 115 IBR IF NOT
6261 006654 000207 RTS PC IEXIT
    
```

```

6268 .SBTTL STATIC DAC CALIBRATION
6269
6270
6271 006656 012706 001100 STATIC: MOV #STACK, SP ILOAD STACK POINTER
6272 006662 004737 001746 JSR PC, LDTRAP ILOAD BUS ADDRESSES
6273 006666 104410 18: CKSWR ITEST FOR CTRL G
6274 006670 017700 172244 MOV #SWR, R0 IREAD SWITCHES
6275 006674 010077 172522 MOV R0, #DAC0 ILOAD DAC #0
6276 006700 010077 172520 MOV R0, #DAC1 ILOAD DAC #1
6277 006704 010077 172516 MOV R0, #DAC2 ILOAD DAC #2
6278 006710 010077 172514 MOV R0, #DAC3 ILOAD DAC #3
6279 006714 000764 RR 15
6280
6281 .SBTTL DYNAMIC DAC CALIBRATION
6282
6283 006716 012706 001100 DYNCAL: MOV #STACK, SP ILOAD STACK POINTER
6284 006722 004737 001746 JSR PC, LDTRAP ILOAD BUS ADDRESSES
6285 006726 104410 18: CKSWR ITEST FOR CTRL G
6286 006730 017700 172204 MOV #SWR, R0 IREAD SWR
6287 006734 004737 006750 JSR PC, 105 ILOAD THE SWR VALUE TO ALL DACS
6288 006740 005000 CLR R0 ICLEAR R0
6289 006742 004737 006750 JSR PC, 105 ILOAD ALL DAC'S WITH R
6290 006746 000767 BR 15
6291
6292 006750 010077 172446 105: MOV R0, #DAC0 ILOAD DAC #0
6293 006754 010077 172444 MOV R0, #DAC1 ILOAD DAC #1
6294 006760 010077 172442 MOV R0, #DAC2 ILOAD DAC #2
6295 006764 010077 172440 MOV R0, #DAC3 ILOAD DAC #3
6296 006770 012700 000020 MOV #20, R0 ILOAD DELAY COUNTER
6297 006774 005300 115: DEC R0 IDELAY
6298 006776 100376 BPL 115 IWAIT
6299 007000 000207 RTS PC IEXIT
6300
6301 DACBAD: ABASE
6302 007004 000000 BADUNT: 0
6303 007006 000004 NUMBOK: 4,
6304 007010 000001 MASKNM: 8170
6305 007012 000000 TEMP: 0
6306 007014 000000 STEMP: 0
6307 007016 000000 SPREAD: 0
6308 007020 000000 WFTST: 0
6309 007022 167774 FILZ: 167774
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
6327 007040 005015 040042 053101 TITLE: .ASCIZ <15><12><12>'AAV11 DIAGNOSTIC TEST, (MAINDEC-11-DVAAA-AP)'<<15><12>
      007046 030461 042040 040511
      007054 047107 051517 040524
      007062 020103 042524 052123
      007070 020054 046450 040501
      007076 042116 041505 030455
      007104 026461 053104 040501
      007112 026501 030101 006451
      007120 000012
6328 007122 052502 020123 040524 EM11 .ASCIZ /RUS TIME-OUT WHEN REFERENCING A DAC ADDRESS/
      007130 042515 047455 052125
      007136 053440 042510 020116
      007144 042522 042506 042522
      007152 041516 047111 000107
      007160 020101 040504 020103
      007166 042101 051104 051505
      007174 000123
6329 007176 040504 030103 051040 EM21 .ASCIZ /DAC0 REGISTER IN ERROR/
      007204 043505 051511 042524
      007212 020122 047111 042440
      007220 051122 051117 000
6330 007225 104 041501 020061 EM31 .ASCIZ /DAC1 REGISTER IN ERROR/
      007232 042522 044507 052123
      007240 051105 044440 000116
      007246 051105 047522 000122
6331 007254 040504 031103 051040 EM41 .ASCIZ /DACR REGISTER IN ERROR/
      007262 043505 051511 042524
      007270 020122 047111 042440
      007276 051122 051117 000
6332 007303 104 041501 020063 EM51 .ASCIZ /DAC3 REGISTER IN ERROR/
      007310 042522 044507 052123
      007316 051105 044440 020116
      007324 051105 047522 000122
6333 007332 042523 042514 052103 EM61 .ASCIZ /SELECTED DAC OFFSET POT WAS ADJUSTED INCORRECTLY/
      007340 042105 042440 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 .ASCIZ /SELECTED DAC GAIN POT WAS ADJUSTED INCORRECTLY/
      007420 042524 020104 040504
      007426 020103 040507 047111
      007434 050040 052117 053440
      007442 051501 040440 040104
      007450 051525 042524 020104
      007456 047111 047503 051122
      007464 041505 046124 000131
6335 007472 042523 042514 052103 EM101 .ASCIZ /SELECTED DAC HAS A LINEARITY PROBLEM/
      007500 042105 042040 041501
      007506 044040 051501 040440

```

```

        .ASCII MESSAGES
6336 007514 046040 047111 040505 EM111 .ASCIZ /+15 VOLT SUPPLY IS INCORRECT/
      007522 044522 050524 050040
      007530 047522 046102 046505
      007536 000
      007537 053 032461 053040
      007544 046117 020124 052523
      007552 050120 050514 044440
      007560 020123 047111 047503
      007566 051122 041405 000124
6337 007574 030455 020065 047526 EM121 .ASCIZ /-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 .ASCIZ /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 .ASCIZ <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 046124 020105
      007740 047520 003524 000007
6340 007746 040511 050103 042522 L0SPAC1 .ASCIZ / DEPRESS THE "SPACE-BAR" WHEN DONE/
      007754 051523 052040 042510
      007762 021040 050123 041501
      007770 026505 040502 021122
      007776 053440 042510 020116
      010004 047504 042516 000
6341 010011 007 005015 042524 M068W1 .ASCII <7><15><12>/TESTER SW1=2 AND SW1=5 ONLY MUST BE ON/
      010016 052123 051105 051440
      010024 030527 031055 040440
      010032 042116 051440 030527
      010040 032455 047440 046116
      010046 020131 052515 052123
      010054 041040 020105 047117
      010062 006407 051412 031127
6342 010070 031055 051440 .ASCII <7><15><12>/SW2=2 SW2=4 AND SW2=5 MUST BE ON/
      010076 032055 040440 042116
      010104 051440 031127 032455
      010112 046440 051525 020124
      010120 042502 047440 116
6343 010125 015 003412 047503 .ASCIZ <15><12><7>/CONNECT AAV11 TO J09 OF TESTER ONLY/<15><12>
      010132 047116 041505 020124
      010140 040501 030526 020061
      010146 047524 045040 034460
      010154 047440 020106 042524
      010162 052123 051105 047440
      010170 046116 006531 000012
6344 010176 021440 042440 051122 ERR0Y1 .ASCIZ / # ERRORS/
      010204 051117 000123

```

```

6345 010210 041040 042101 052440 MMSG01 .ASCIZ / RAD UNITS /
      010216 044516 051524 000040
6346 010224 015 P12
6347 010226 051120 043517 040522 FOUND01 .BYTE 15,12
      010234 020115 042504 042524 .ASCIZ /PROGRAM DETECTED /
      010242 052103 042105 000040
6348 010250 034050 004451 040501 FOUND02 .ASCIZ / (8) AAV11(8) /
      010256 030526 024061 024523
      010264 020040 000
6349 010267 105 051122 041520 DM1: .ASCIZ /ERRPC BUSADR EXPECT WAS/
      010274 041011 051525 042101
      010302 020122 042440 050130
      010310 041505 020124 020040
      010316 040527 000123
6350 010322 051105 050122 004503 DM2: .ASCIZ /ERRPC BUSADR/
      010330 052502 040523 051104
      010336 000
6351 010337 105 051122 041520 DM6: .ASCIZ /ERRPC BUSADR EXPECT WAS SPREAD/
      010344 041011 051525 042101
      010352 004522 054105 042520
      010360 052103 053411 051501
      010366 051411 051120 040505
      010374 000104
6355 010376 005015 040440 045104 ADJR34: .ASCIZ <15><12>/ ADJUST R34 FOR A NULL /
      (1) 010404 051525 020124 031522
      (1) 010412 020064 047506 020122
      (1) 010420 020101 052516 046114
      (1) 010426 020040 000
6356 010431 015 020012 042101 ADJR35: .ASCIZ <15><12>/ ADJUST R35 FOR A NULL /
      (1) 010436 052512 052123 051040
      (1) 010444 032463 043040 051117
      (1) 010452 040440 047040 046125
      (1) 010460 020114 000040
6357 010464 005015 040440 045104 ADJR36: .ASCIZ <15><12>/ ADJUST R36 FOR A NULL /
      (1) 010472 051525 020124 031522
      (1) 010500 020066 047506 020122
      (1) 010506 020101 052516 046114
      (1) 010514 020040 000
6358 010517 015 020012 042101 ADJR37: .ASCIZ <15><12>/ ADJUST R37 FOR A NULL /
      (1) 010524 052512 052123 051040
      (1) 010532 033063 043040 051117
      (1) 010540 040440 047040 046125
      (1) 010546 020114 000040
6359 010552 005015 040440 045104 ADJR46: .ASCIZ <15><12>/ ADJUST R46 FOR A NULL /
      (1) 010560 051525 020124 032122
      (1) 010566 020066 047506 020122
      (1) 010574 020101 052516 046114
      (1) 010602 020040 000
6360 010605 015 020012 042101 ADJR47: .ASCIZ <15><12>/ ADJUST R47 FOR A NULL /
      (1) 010612 052512 052123 051040
      (1) 010620 033064 043040 051117
      (1) 010626 040440 047040 046125
      (1) 010634 020114 000040
6361 010640 005015 040440 045104 ADJR48: .ASCIZ <15><12>/ ADJUST R48 FOR A NULL /
      (1) 010646 051525 020124 032122
      (1) 010654 020070 047506 020122

```

```

(1) 010662 020101 052516 046114
(1) 010670 020040 000
6362 010673 015 020012 042101 ADJR49: .ASCIZ <15><12>/ ADJUST R49 FOR A NULL /
      (1) 010700 052512 052123 051040
      (1) 010706 034064 043040 051117
      (1) 010714 040440 047040 046125
      (1) 010722 020114 000040
6363
6367 010726 005015 042523 042514 SELD0: .ASCIZ <15><12>/SELECT DAC0/
      (1) 010734 052103 042040 041501
      (1) 010742 000060
6368 010744 005015 042523 042514 SELD1: .ASCIZ <15><12>/SELECT DAC1/
      (1) 010752 052103 042040 041501
      (1) 010760 000061
6369 010762 005015 042523 042514 SELD2: .ASCIZ <15><12>/SELECT DAC2/
      (1) 010770 052103 042040 041501
      (1) 010776 000062
6370 011000 005015 042523 042514 SELD3: .ASCIZ <15><12>/SELECT DAC3/
      (1) 011006 052103 042040 041501
      (1) 011014 000063
6371 011016 005015 042101 052512 TRYAGN: .ASCIZ <15><12>/ADJUST THAT SAME POT AGAIN PLEASE/<15><12>
      011024 052123 052040 040510
      011032 020124 040523 042515
      011040 050040 052117 040440
      011046 040507 047111 050040
      011054 042514 051501 006505
      011062 000012
6372 000001 STX=1
6373 000003 ETX=3
6374 011064 001 N51200: .BYTE STX
6375 011065 116 030465 030062 .ASCIZ /N512000V/
      011072 030060 126
6376 011075 003 000 .BYTE ETX,0
6377 011077 001 .BYTE STX
6378 011100 032520 030461 032467 .ASCIZ /P511750V/
      011106 053060
6379 011110 003 000 .BYTE ETX,0
6380 .EVEN
6381 011112 001116 001126 000000 DT1: SERRPC,$DDAT,0
6382 011120 001116 001422 001124 DT2: SERRPC,DAC0,$GDDAT,$SDDAT,0
      011126 001126 000000
6383 011132 001116 001424 001124 DT3: SERRPC,DAC1,$GDDAT,$SDDAT,0
      011140 001126 000000
6384 011144 001116 001426 001124 DT4: SERRPC,DAC2,$GDDAT,$SDDAT,0
      011152 001126 000000
6385 011156 001116 001430 001124 DT5: SERRPC,DAC3,$GDDAT,$SDDAT,0
      011164 001126 000000
6386 011170 001116 007002 001124 DT6: SERRPC,DACRAD,$GDDAT,$SDDAT,SPREAD,0
      011176 001126 007016 000000
6387 011204 000000 000000 000000 DF0: 0,0,0,0,0,0,0,0
      011212 000000 000000 000000
      011220 000000 000000
6388
6389 .SBTTL BINARY TO ASCII AND TYPE ROUTINE
(1)
(2)
*****

```



```

(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 3R //SKIP TYPEOUT IF NO POINTER
(1) 012252 104401 TYPE //TYPE THE "ERROR MESSAGE"
(1) 012254 000000 // "ERROR MESSAGE" POINTER GOES HERE
(1) 012256 104401 001171 .SCRLF // "CARRIAGE RETURN" & "LINE FEED"
(1) 012262 012037 012272 3S: MOV (R0)+,4S //PICKUP "DATA HEADER" POINTER
(1) 012266 001404 REQ 5S //SKIP TYPEOUT IF 0
(1) 012270 104401 TYPE //TYPE THE "DATA HEADER"
(1) 012272 000000 // "DATA HEADER" POINTER GOES HERE
(1) 012274 104401 001171 .SCRLF // "CARRIAGE RETURN" & "LINE FEED"
(1) 012300 011000 5S: MOV (R0),R0 //PICKUP "DATA TABLE" POINTER
(1) 012302 001004 BNE 7S //GO TYPE THE DATA
(1) 012304 012600 6S: MOV (SP)+,R0 //RESTORE R0
(1) 012306 104401 001171 .SCRLF // "CARRIAGE RETURN" & "LINE FEED"
(1) 012312 000207 RTS PC //RETURN
(2) 012314 7S: MOV # (R0)+,-(SP) //SAVE # (R0)+ FOR TYPEOUT
(2) 012316 104402 TYPOC //GO TYPE--OCTAL ASCII(ALL DIGITS)
(1) 012320 005710 TST (R0) //IS THERE ANOTHER NUMBER?
(1) 012322 001770 BEQ 6S //BR IF NO
(1) 012324 104401 012332 TYPE ,AS //TYPE TWO(2) SPACES
(1) 012330 000771 BR 7R //LOOP
(1) 012332 000040 000 8S: .ASCIZ / / //TWO(2) SPACES
(1) 012336 012336 .EVEN
6394 .SBTTL POWER DOWN AND UP ROUTINES
(1)
(2)
//*****
//POWER DOWN ROUTINE
(1) 012336 012737 012502 000024 SPWRDN: MOV #SILLUP,#PWRVEC //SET FOR FAST UP
(1) 012344 012737 000340 000026 MOV #340,#PWRVEC+2 //PRI017
(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)
//*****
//POWER UP ROUTINE
(1) 012410 012737 012502 000024 SPWRUP: MOV #SILLUP,#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) 012434 001375 BNE 1S //OP WORD
(1) 012440 012677 166500 MOV (SP)+,#SWR //POP STACK INTO #SWR
(1) 012442 012605 MOV (SP)+,R5 //POP STACK INTO R5
(1) 012444 012604 MOV (SP)+,R4 //POP STACK INTO R4
(1) 012446 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 //PRI017
(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 REGIN //RESTART ADDRESS
(1) 012500 000002 RTI
(1) 012502 000000 SILLUP: 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 PWRMSG: .ASCIZ <15><12>/RESTARTING AFTER A POWER FAILURE/<15><12><12>
(1) 012510 005015 042522 052123
(1) 012516 051101 044524 043516
(1) 012524 040440 057106 051105
(1) 012532 040440 050040 053517
(1) 012540 051105 043040 044501
(1) 012546 052514 042522 005015
(1) 012554 000012
6396 .EVEN
6397 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
6398
(1)
(2)
//*****
//THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
//OCTAL (ASCII) NUMBER AND TYPE IT.
//STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
//CALL:
// MOV NUM,-(SP) //NUMBER TO BE TYPED
// TYPOS //CALL FOR TYPEOUT
// .BYTE N //N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
// .BYTE M //M=1 OR 0
// //TYPE LEADING ZEROS
// //SUPPRESS LEADING ZEROS
(1)
(1)
//STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
//STYPOS OR STYPOC
//CALL:
// MOV NUM,-(SP) //NUMBER TO BE TYPED
// TYPON //CALL FOR TYPEOUT
(1)
(1)
//STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
//CALL:
// MOV NUM,-(SP) //NUMBER TO BE TYPED
// TYPOC //CALL FOR TYPEOUT
(1)
(1)
(1) 012556 017644 000000 STYPOS: MOV # (SP)-,(SP) //PICKUP THE MODE
(1) 012562 116637 000001 013001 MOVB 1(SP),#SFILL //LOAD ZERO FILL SWITCH
(1) 012570 112637 013003 MOVB (SP)+,#OHMODE+1 //NUMBER OF DIGITS TO TYPE
(1) 012574 002716 000002 ADD #2,(SP) //ADJUST RETURN ADDRESS
(1) 012600 000000 BR STYPON
(1) 012602 112737 000001 013001 STYPOC: MOVB #1,#SFILL //SET THE ZERO FILL SWITCH
(1) 012610 112737 000006 013003 MOVB #6,#OHMODE+1 //SET FOR SIX(6) DIGITS
(1) 012616 112737 000005 013000 STYPON: MOVB #5,#OCNT //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 013002 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 MSR
(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 012776 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
0399 .SRTTL 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 001495 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 $CHARCNT //DO NOT COUNT AS A COUNT
(1) 013170 000770 BR 70 //LOOP
(1)
(1) //HORIZONTAL TAB PROCESSOR
(1) 013172 112716 000040 80: MOV#R #0,(SP) //REPLACE TAB WITH SPACE
(1) 013176 004737 013216 90: JSR PC,STYPEC //TYPE A SPACE
(1) 013202 132737 000007 013262 BIT# #7,$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) 013232 122766 000015 000002 CMPB #CR,2(SP) ;;IS CHARACTER A CARRIAGE RETURN?  
(1) 013240 001003 BNE BRNCH IF NO  
(1) 013242 105037 013262 CLR# SCHARCNT ;;YES--CLEAR CHARACTER COUNT  
(1) 013246 000006 BR STYPEX ;;EXIT  
(1) 013250 122766 000012 000002 101 CMPB #LF,2(SP) ;;IS CHARACTER A LINE FEED?  
(1) 013256 001402 BEQ STYPEX ;;BRANCH IF YES  
(1) 013260 105227 INCR (PC)+ ;;COUNT THE CHARACTER  
(1) 013262 000000 SCHARCNT,WORD 0 ;;CHARACTER COUNT STORAGE  
(1) 013264 000207 STYPEX: RTS PC  
(1) 6400 .SBTTL TTY INPUT ROUTINE  
(1) ;;*****  
(2) .ENARL LSR  
(1) ;;*****  
(1) ;;SOFTWARE SWITCH REGISTER CHANGE ROUTINE.  
(1) ;;ROUTINE IS ENTERED FROM THE TRAP HANDLER, AND WILL  
(1) ;;SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP CALL  
(1) ;;WHEN OPERATING IN TTY FLAG MODE.  
(1) 013266 022737 000176 001140 SCKSWR: CMP #SWREG,SWR ;;IS THE SOFT-SWR SELECTED?  
(1) 013274 001074 BNE BRNCH IF NO  
(1) 013276 105777 165642 TSTR #STKS ;;CHAR THERE?  
(1) 013302 100071 BPL LSR ;;IF NO, DON'T WAIT AROUND  
(1) 013304 117746 165636 MOV# #STKB,-(SP) ;;SAVE THE CHAR  
(1) 013310 042716 177600 RIC #C177,(SP) ;;STRIP-OFF THE ASCII  
(1) 013314 022726 000007 CMP #7,(SP)+ ;;IS IT A CONTROL G?  
(1) 013320 001062 BNE LSR ;;NO, RETURN TO USER  
(1) 013322 123727 001134 000001 CMPL #AUTOR,#1 ;;ARE WE RUNNING IN AUTO-MODE?  
(1) 013330 001456 BEQ LSR ;;BRANCH IF YES  
(1) 013332 104401 014013 SGTSWR: TYPE ,SCNTLG ;;ECHO THE CONTROL-G (^G)  
(1) 013336 104401 014020 TYPE ,MSWR ;;TYPE CURRENT CONTENTS  
(2) 013342 013746 000176 MOV SWREG,-(SP) ;;SAVE SWREG FOR TYPEOUT  
(2) 013346 104402 TYPEOC TYPEOC ;;GO TYPE--OCTAL ASCII(ALL DIGITS)  
(1) 013350 104401 014031 TYPE ,SNEW ;;PROMPT FOR NEW SWR  
(1) 013354 005046 CLR -(SP) ;;CLEAR COUNTER  
(1) 013356 005046 CLR -(SP) ;;THE NEW SWR  
(1) 013360 105777 165560 7S: TSTR #STKS ;;CHAR THERE?  
(1) 013364 100375 RPL 7S ;;IF NOT TRY AGAIN  
(1) 013366 117746 165554 MOV# #STKB,-(SP) ;;PICK UP CHAR  
(1) 013372 042716 177600 RIC #C177,(SP) ;;MAKE IT 7-BIT ASCII  
(1) 013376 021627 000025 9S: CMP (SP),#25 ;;IS IT A CONTROL-U?  
(1) 013402 001005 BNE BRNCH IF NOT  
(1) 013404 104401 014006 TYPE ,SCNTLU ;;YES, ECHO CONTROL-U (^U)  
(1) 013410 062706 000006 20S: ADD #6,SP ;;IGNORE PREVIOUS INPUT  
(1) 013414 000757 RR 19S ;;LET'S TRY IT AGAIN  
(1) 013416 021627 000015 10S: CMP (SP),#15 ;;IS IT A <CR>?  
(1) 013422 001022 BNE BRNCH IF NO  
(1) 013424 005766 000004 TST 4(SP) ;;YES, IS IT THE FIRST CHAR?
```

```
(1) 013430 001403 BEQ LSR ;;BRANCH IF YES  
(1) 013432 016677 000002 165500 MOV 2(SP),#SWR ;;SAVE NEW SWR  
(1) 013440 062706 000006 11S: ADD #6,SP ;;CLEAR UP STACK  
(1) 013444 104401 001171 14S: TYPE ,SCRLF ;;CLEAR <CR> AND <LF>  
(1) 013450 123727 001135 000001 15S: CMPB $INTAG,#1 ;;RE-ENABLE TTY KBD INTERRUPTS?  
(1) 013456 001003 BNE LSR ;;BRANCH IF NOT  
(1) 013460 012777 000100 165456 MOV #100,#STKS ;;RE-ENABLE TTY KBD INTERRUPTS  
(1) 013466 000002 RTI ;;RETURN  
(1) 013470 004737 013216 16S: JSR PC,STYPEC ;;ECHO CHAR  
(1) 013474 021627 000060 CMP (SP),#60 ;;CHAR < 6?  
(1) 013500 002420 BLT 10S ;;BRANCH IF YES  
(1) 013502 021627 000067 CMP (SP),#67 ;;CHAR < 7?  
(1) 013506 003015 RGT 10S ;;BRANCH IF YES  
(1) 013510 042726 000060 BIC #60,(SP)+ ;;STRIP-OFF ASCII  
(1) 013514 005766 000002 TST 2(SP) ;;IS THIS THE FIRST CHAR  
(1) 013520 001403 BEQ LSR ;;BRANCH IF YES  
(1) 013522 006316 ASL (SP) ;;NO, SHIFT PRESENT  
(1) 013524 006316 ASL (SP) ;; CHAR OVER TO MAKE  
(1) 013526 006316 ASL (SP) ;; ROOM FOR NEW ONE.  
(1) 013530 005266 000002 17S: INC 2(SP) ;;KEEP COUNT OF CHAR  
(1) 013534 056616 177776 RIS -2(SP),(SP) ;;SET IN NEW CHAR  
(1) 013540 000707 BR 7S ;;GET THE NEXT ONE  
(1) 013542 104401 001170 18S: TYPE ,SQUES ;;TYPE 1<CR>=<LF>  
(1) 013546 000720 BR 20S ;;SIMULATE CONTROL-U  
(1) .DSARL LSR  
(1) ;;*****  
(2) ;;THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY  
(1) ;;CALL:  
(1) ;;  
(1) ;;  
(1) ;;  
(1) ;;  
(1) ;;  
(1) ;;  
(1) 013550 011646 SRDCHR: MOV (SP),-(SP) ;;PUSH DOWN THE PC  
(1) 013552 016666 MOV 4(SP),2(SP) ;;SAVE THE PS  
(1) 013560 105777 165360 1S: TSTR #STKS ;;WAIT FOR  
(1) 013564 100375 BPL LSR ;;A CHARACTER  
(1) 013566 117746 165354 000004 MOV# #STKB,4(SP) ;;READ THE TTY  
(1) 013574 042766 177600 000004 RIC #C177,4(SP) ;;GET RID OF JUNK IF ANY  
(1) 013602 026627 000004 000023 CMP 4(SP),#23 ;;IS IT A CONTROL-S?  
(1) 013610 001013 BNE LSR ;;BRANCH IF NO  
(1) 013612 105777 165326 2S: TSTR #STKS ;;WAIT FOR A CHARACTER  
(1) 013616 100375 RPL 2S ;;LOOP UNTIL ITS THERE  
(1) 013620 117746 165322 MOV# #STKB,-(SP) ;;GET CHARACTER  
(1) 013624 042716 177600 RIC #C177,(SP) ;;MAKE IT 7-BIT ASCII  
(1) 013630 022627 000021 CMP (SP)+,#21 ;;IS IT A CONTROL-G?  
(1) 013634 001366 BNE LSR ;;IF NOT DISCARD IT  
(1) 013636 000750 BR LSR ;;YES, RESUME  
(1) 013640 026627 000004 000140 3S: CMP 4(SP),#140 ;;IS IT UPPER CASE?  
(1) 013646 002407 BLT 4S ;;BRANCH IF YES  
(1) 013650 026627 000004 000175 4S: CMP 4(SP),#175 ;;IS IT A SPECIAL CHAR?  
(1) 013656 003003 BGT 4S ;;BRANCH IF YES  
(1) 013660 042766 000004 000004 BIC #40,4(SP) ;;MAKE IT UPPER CASE  
(1) 013666 000002 RTI ;;GO BACK TO USER
```

```

(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 013670 010346      SRDLIN: MOV    R3,-(SP)      ;;SAVE R3
(1) 013672 012703 013776 18:  MOV    #ATTYIN,R3      ;;GET ADDRESS
(1) 013676 022703 014006 20:  CMP    #ATTYIN+8,,R3      ;;BUFFER FULL?
(1) 013702 010405      BLOS    48                   ;;BR IF YES
(1) 013704 044411      RDCMR   ;;GD READ ONE CHARACTER FROM THE TTY
(1) 013706 112613      MOV    (SP)+,(R3)          ;;GET CHARACTER
(1) 013710 122713 000177 100:  CMPR   #177,(R3)          ;;IS IT A RUBOUT
(1) 013714 001003      BNE    38                   ;;SKIP IF NOT
(1) 013716 104401 001170 48:  TYPE   ,SQUES              ;;TYPE A "Q"
(1) 013722 000763      BR     18                   ;;CLEAR THE BUFFER AND LOOP
(1) 013724 111337 013774 30:  MOV    (R3),98             ;;ECHO THE CHARACTER
(1) 013730 104401 013774      TYPE   ,98
(1) 013734 122723 000015      CMPS   #15,(R3)+          ;;CHECK FOR RETURN
(1) 013740 001356      RNE    R3                   ;;LOOP IF NOT RETURN
(1) 013742 105063 177777      CLRR   -(R3)              ;;CLEAR RETURN (THE 15)
(1) 013746 104401 001172      TYPE   ,SLF              ;;TYPE A LINE FEED
(1) 013752 012603      MOV    (SP)+,R3           ;;RESTORE R3
(1) 013754 011646      MOV    (SP)-,(SP)         ;;ADJUST THE STACK AND PUT ADDRESS OF THE
(1) 013756 016666 000004 000002  MOV    4(SP),2(SP)        ;; FIRST ASCII CHARACTER ON IT
(1) 013764 012766 013776 000004  MOV    #ATTYIN,4(SP)
(1) 013772 000002      RTI
(1) 013774 000      98:  .BYTE 0                      ;;STORAGE FOR ASCII CHAR, TO TYPE
(1) 013775 000      .BYTE 0                      ;;TERMINATOR
(1) 013776 000010      .ALKA A                      ;;RESERVE 8 BYTES FOR TTY INPUT
(1) 014006 052536 005015 000  SCNTLUI .ASCIZ /U/<15><12>     ;;CONTROL "U"
(1) 014013 136 006507 000012  SCNTLGI .ASCIZ /G/<15><12>     ;;CONTROL "G"
(1) 014020 005019 053523 020122  SMRRI .ASCIZ <15><12>/SMR = /
(1) 014026 020075 000
(1) 014031 040 047000 053505  SMNEWI .ASCIZ / NEW = /
(1) 014036 036440 000040
640:
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 014042 011646      SRDOCT: MOV    (SP)-,(SP)     ;;PROVIDE SPACE FOR THE
(1) 014044 016666 000004 000002  MOV    4(SP),2(SP)         ;;INPUT NUMBER
(1) 014052 010046      MOV    R0,-(SP)           ;;PUSH R0 ON STACK
(1) 014054 010146      MOV    R1,-(SP)           ;;PUSH R1 ON STACK
(1) 014056 010246      MOV    R2,-(SP)           ;;PUSH R2 ON STACK
(1) 014060 104412 18:  RDLIN  ;;READ AN ASCII LINE
(1) 014062 012600      MOV    (SP)+,R0           ;;GET ADDRESS OF 1ST CHARACTER
(1) 014064 005001      CLR    R1                  ;;CLEAR DATA WORD
(1) 014066 005002      CLR    R2
    
```

```

(1) 014070 112046      29:  MOVR   (R0)+,(SP)          ;;PICKUP THIS CHARACTER
(1) 014072 001412      REQ    38                   ;;IF ZERO GET OUT
(1) 014074 006301      ASL    R1                    ;;*2
(1) 014076 006102      ROL    R2                    ;;
(1) 014100 006301      ASL    R1                    ;;*4
(1) 014102 006102      ROL    R2                    ;;
(1) 014104 006301      ASL    R1                    ;;*8
(1) 014106 006102      ROL    R2                    ;;
(1) 014110 042716 177770      BIC    #C7,(SP)           ;;STRIP THE ASCII JUNK
(1) 014114 062601      ADD    (SP)+,R1            ;;AND IN THIS DIGIT
(1) 014116 000764      BR     28                   ;;LOOP
(1) 014120 005726      30:  TST    (SP)+              ;;CLEAN TERMINATOR FROM STACK
(1) 014122 010166 000012      MOV    R1,12(SP)          ;;SAVE THE RESULT
(1) 014126 010237 014142      MOV    R2,SHIOCT          ;;
(1) 014132 012602      MOV    (SP)+,R2           ;;POP STACK INTO R2
(1) 014134 012601      MOV    (SP)+,R1           ;;POP STACK INTO R1
(1) 014136 012600      MOV    (SP)+,R0           ;;POP STACK INTO R0
(1) 014140 000002      RTI                        ;;RETURN
(1) 014142 000000      SHIOCT: .WORD 0              ;;HIGH ORDER BITS GO HERE
(1)
(2)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1)
(1) 014144 112737 000001 014410  SATY1: MOVR   #1,SFFLG          ;;TO REPORT FATAL ERROR
(1) 014152 112737 000001 014406  SATY3: MOVR   #1,SMFLG          ;;TO TYPE A MESSAGE
(1) 014160 000403      RR     #SATYC
(1) 014162 112737 000001 014410  SATY4: MOVA   #1,SFFLG          ;;TO ONLY REPORT FATAL ERROR
(2) 014170
(3) 014170 010046      MOV    R0,-(SP)           ;;PUSH R0 ON STACK
(3) 014172 010146      MOV    R1,-(SP)           ;;PUSH R1 ON STACK
(1) 014174 105737 014406      TSTB   SMFLG              ;;SHOULD TYPE A MESSAGE?
(1) 014200 001450      REQ    58                   ;;IF NOT: BR
(1) 014202 122737 000001 001214  CMPR   #APTENV,SENV          ;;OPERATING UNDER APT?
(1) 014210 001031      BNE    38                   ;;IF NOT: BR
(1) 014212 132737 000100 001215  BITB   #APTSPOOL,SENVM      ;;SHOULD SPOOL MESSAGES?
(1) 014220 001425      BEQ    38                   ;;IF NOT: BR
(1) 014222 017600 000004      MOV    #4(SP),R0          ;;GET MESSAGE ADDR.
(1) 014224 062766 000002 000004  ADD    #2,4(SP)            ;;BUMP RETURN ADDR.
(1) 014234 005737 001174      TST    #MSGTYPE           ;;SEE IF DONE W/ LAST XMISSION?
(1) 014240 001375      BNE    18                   ;;IF NOT: WAIT
(1) 014242 010037 001210      MOV    R0,#MSGAD          ;;PUT ADDR IN MAILBOX
(1) 014246 105720 28:  TSTB   (R0)+              ;;FIND END OF MESSAGE
(1) 014250 001376      RNE    28                   ;;
(1) 014252 163700 001210      BUR    #MSGAD,R0          ;;SUB START OF MESSAGE
(1) 014256 006200      ASR    R0                    ;;GET MESSAGE LNTH IN WORDS
(1) 014260 010037 001212      MOV    R0,#MSGLG          ;;PUT LENGTH IN MAILBOX
(1) 014264 012737 000004 001174  MOV    #4,#MSGTYPE         ;;TELL APT TO TAKE MSG.
(1) 014272 000613      RR     #5
(1) 014274 017637 000004 014320 30:  MOV    #4(SP),48          ;;PUT MSG ADDR IN JSR LINKAGE
(1) 014302 062766 000002 000004  ADD    #2,4(SP)            ;;BUMP RETURN ADDRESS
(3) 014310 013746 177776      MOV    177776,-(SP)       ;;PUSH 177776 ON STACK
(1) 014314 004737 013004      JSR    PC,9TYPE           ;;CALL TYPE MACRO
(1) 014320 000000      .WORD 0
(1) 014322
(1) 014324 105737 014410 100:  TSTB   SFFLG              ;;SHOULD REPORT FATAL ERROR?
(1) 014326 001410      BEQ    128                  ;;IF NOT: BR
(1) 014330 005737 001214      TST    SENV                ;;RUNNING UNDER APT?
    
```

```

(1) 014334 001413          BEQ      129          ;;IF NOT: BR
(1) 014336 005737 001174    118:    TST      SMSTGTYPE  ;;FINISHED LAST MESSAGE?
(1) 014342 001375          BNE      118          ;;IF NOT: WAIT
(1) 014344 017637 000004 001176  MOV      #4(SP),SFATAL ;;GET ERROR #
(1) 014352 062766 000002 000204  ADD      #2,4(SP)      ;;BUMP RETURN ADDR.
(1) 014360 005237 001174    INC      SMSTGTYPE     ;;TELL APT TO TAKE ERROR
(1) 014364 105037 014410    129:    CLRB    SPFLG        ;;CLEAR FATAL FLAG
(1) 014370 105037 014407    CLRB    SLFLG        ;;CLEAR LOG FLAG
(1) 014374 105037 014406    CLRB    SMFLG        ;;CLEAR MESSAGE FLAG
(3) 014400 012601          MOV      (SP)+,R1     ;;POP STACK INTO R1
(3) 014402 012600          MOV      (SP)+,R0     ;;POP STACK INTO R0
(1) 014404 000207          RTS      PC           ;;RETURN
(1) 014406 000          SMFLG: .BYTE 0       ;;MESSAGE FLAG
(1) 014407 000          SLFLG: .BYTE 0       ;;LOG FLAG
(1) 014410 000          SPFLG: .BYTE 0       ;;FATAL FLAG
(1)          014412          .EVEN
(1)          000200          APTSIZE=200
(1)          000001          APTENV=001
(1)          000100          APTSPool=100
(1)          000004          APTCSUP=004
6405

```

```

6405          .SBTTL TRAP DECODER
(1)
(2)
(1)          ;;*****
(1)          ;;THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
(1)          ;;AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
(1)          ;;OF THE DESIRED ROUTINE, THEN USING THE ADDRESS OBTAINED IT WILL
(1)          ;;GO TO THAT ROUTINE.
(1) 014412 010046          STRAP:  MOV      R0,-(SP)      ;;SAVE R0
(1) 014414 016600 000002    MOV      2(SP),R0          ;;GET TRAP ADDRESS
(1) 014420 005740          TST      -(R0)            ;;BACKUP BY 2
(1) 014422 111000          MOV8     (R0),R0          ;;GET RIGHT BYTE OF TRAP
(1) 014424 006300          ASL     R0                ;;POSITION FOR INDEXING
(1) 014426 016000 014446    MOV      STRPAD(R0),R0     ;;INDEX TO TABLE
(1) 014432 000200          RTS      R0              ;;GO TO ROUTINE
(1)
(1)
(1)          ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
(1) 014434 011666          STRAPP: MOV      (SP),-(SP)  ;;MOVE THE PC DOWN
(1) 014436 016666 000004 000002  MOV      4(SP),2(SP)      ;;MOVE THE PSW DOWN
(1) 014444 000002          RTI                    ;;RESTORE THE PSW
(1)
(3)          .SBTTL TRAP TABLE
(3)
(3)          ;;THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
(3)          ;;BY THE "TRAP" INSTRUCTION.
(3)
(3)          |
(3)          | ROUTINE
(3)          | -----
(3) 014446 014434          STRAPP: .WORD    STRAPP
(3) 014450 013004          STYPE   STYPE      ;;CALL=TYPE      TRAP+1(104401) TTY TYPEOUT ROUTINE
(3) 014452 012602          STYPOC STYPOC     ;;CALL=TYPOC   TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
(3) 014454 012556          STYPOS STYPOS     ;;CALL=TYPOS   TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
(3) 014456 012610          STYPOB STYPOB     ;;CALL=TYPOB   TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LABY CALL)
(3) 014460 011300          STYPOD STYPOD     ;;CALL=TYPOD   TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)
(3) 014462 011224          STYPOB STYPOB     ;;CALL=TYPOB   TRAP+6(104406) TYPE BINARY (ASCII) NUMBER
(1)
(3) 014464 013336          BGTSHR  ;;CALL=BGTSHR   TRAP+7(104407) GET SOFT-SWR SETTING
(1)
(3) 014466 013266          BCKSHR  ;;CALL=BCKSHR   TRAP+8(104410) TEST FOR CHANGE IN SOFT-SWR
(3) 014470 013550          BRDCHR  ;;CALL=BRDCHR   TRAP+9(104411) TTY TYPEIN CHARACTER ROUTINE
(3) 014472 013670          BRDLIN  ;;CALL=BRDLIN   TRAP+10(104412) TTY TYPEIN STRING ROUTINE
(3) 014474 014042          BRDOCT  ;;CALL=BRDOCT   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#	5012	5021	5039	5047	5067	5076	5097	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	= 006520	6060	6049	6093	6119	6142	6224#												
CONVRT	= 006302	6037	6046	6090	6116	6130	6177#												
CR	= 000015	5551#	6399																
CRLF	= 000200	5551#	6399																
CSPACE	= 000422	6080	6114	6202#															
DACBAD	= 007002	6077*	6106*	6132*	6301#	6306													
DAC0	= 001422	6053*	6110*	6185	6397*	6398	6005*	6006	6013*	6014	6021*	6022*	6011*	6016					
DAC1	= 001424	6040*	6043	6062*	6094	6252	6275*	6292*	6302	6013*	6014	6021*	6022*	6011*	6016				
DAC2	= 001426	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540	60540
DAC3	= 001430	6023	6059	6062	6084	6085	6256	6277*	6294*	6304	6013*	6014	6021*	6022*	6011*	6016			
DDISP	= 177570	5551#	5574	5696															
DF0	= 011204	5502	5508		5600	5606	5612	5619	5625	5631	5637	5643	6307#						

S007	000200	5551#
S008	000400	5551#
S009	001000	5551#
S01	000002	5551#
S010	002000	5551#
S011	004000	5551#
S012	010000	5551#
S013	020000	5551#
S014	040000	5551#
S015	100000	5551#
S02	000004	5551#
S03	000010	5551#
S04	000020	5551#
S05	000040	5551#
S06	000100	5551#
S07	000200	5551#
S08	000400	5551#
S09	001000	5551#
TBITVE	000014	5551#
TEMP	007012	5492# 5730 5969# 6305#
TEATER	001432	5566 5686#
TITLE	007040	5739 6327#
TKVEC	000060	5551#
TPVEC	000064	5551#
TRAPVE	000034	5551# 5696#
TRTVEC	000014	5551#
TRYAGN	011016	6097 6123 6371#
TST1	002306	5759 5703# 5988
TST10	002752	5820 5830#
TST11	003010	5835 5838#
TST12	003062	5847#
TST13	003154	5848#
TST14	003250	5850#
TST15	003304	5855 5858#
TST16	003342	5863 5866#
TST17	003414	5876#
TST2	002364	5795#
TST20	003506	5878#
TST21	003602	5880#
TST22	003636	5885 5888#
TST23	003674	5893 5896#
TST24	003746	5907#
TST25	004040	5908#
TST26	004134	5910#
TST27	004316	5936 5939#
TST3	002420	5800 5803#
TST30	004362	5945 5948#
TST31	004426	5954 5958#
TST32	004464	5963 5966#
TST33	004530	5972 5977#
TST34	004624	5980 6015#
TST35	004646	6017 6020#
TST36	004740	6035#
TST37	005002	6041 6044#
TST4	002456	5808 5811#
TST40	005044	6050 6054#

TST41	005076	6054#
TST42	005126	6054#
TST43	005146	6056#
TST44	005200	6056#
TST45	005230	6056#
TST46	005250	6058#
TST47	005302	6058#
TST5	002530	5821#
TST50	005332	6058#
TST51	005352	6060#
TST52	005404	6060#
TST53	005434	6060#
TST6	002622	5822#
TST7	002716	5823#
TYPNB	104406	6070 6405#
TYPD8	104405	6072 6068 6405#
TYPE	104401	5736 5738 5764 5770 6062 6066 6069 6081 6086 6096 6112 6122 6202
TYPOC	104402	6309 6390 6392 6393 6394 6398 6399 6400 6405#
TYPON	104404	6405#
TYPO8	104403	5767 6405#
VADDR	001416	5651# 5748 5982 5983 5984 5985
V144	007034	6039 6048 6314#
V2034	007036	6045 6315#
V5744	007032	6036 6313#
WPT8T	007020	5686# 5691# 5734 5762 6016 6064 6308#
WPT8D	001000	5573#
WASTAT	***** U	6402
WATYC	014170	6402#
WATY1	014144	6402#
WATY3	014152	6399 6402#
WATY4	014162	6392 6402#
WATYOB	001134	5574# 6400
WBA8E	001250	5574# 5713 5743
WBDADR	001122	5574#
WBDDAT	001126	5574# 5743# 5747 5748# 5798# 5799 5806# 5807 5814# 5815 5821# 5822# 5826#
		5827 5833# 5834 5841# 5842 5847# 5848# 5853# 5854 5861# 5862 5869# 5870
		5876# 5878# 5883# 5884 5891# 5892 5899# 5900 5907# 5908# 5916# 5917 5922#
		5923 5928# 5929 5934# 5935 5943# 5944 5952# 5953 5962# 5971# 6025# 6026#
		6027 6193# 6227 6381 6382 6383 6384 6385 6386
WBELL	001164	5574#
WBIN	011276	6389#
WCDW1	001254	5574#
WCHARC	013262	6399#
WCK8WR	013266	6400# 6405
WCHYAG	001100	5574# 5696
WCH3	000000	5574#
WCNTL6	014013	6400#
WCNTLU	014006	6400#
WCPUPD	001222	5574#
WCRLP	001171	6392 6393 6399 6400
WDBLK	011514	6398#
WDEVCT	001204	5574# 5779# 5981#
WDEVH	001252	5574#
WDDAGN	005552	6062#
WDTBL	011504	6398#

SENDAD	025542	5571	6062#																
SENDCT	025510	6062#																	
SENDMG	025561	6062#																	
SENULL	025556	6062#																	
SENV	001214	5574#	5750	6392	6399	6402													
SENVH	001215	5574#	5696	6399	6402														
SEOP	005454	6018	6062#																
SEOPCT	025502	6062#																	
SERFLG	001103	5574#	6391*	6392*															
SERMAX	001115	5574#	5696*	6391*															
SERROR	012000	5696	6392#																
SERRPC	001116	5574#	6381	6382	6383	6384	6385	6386	6392*	6393									
SERRTB	001256	5574#	6393																
SERRTY	012202	6392	6393#																
SERTYL	001112	5574#	6067	6392*															
SESCAP	001162	5574#	5696*	6391*	6392														
SETABL	001214	5574#																	
SEYEND	001256	5573	5574#																
SPATAL	001176	5574#	6402*																
SFFLG	014410	6402#*																	
SFILLC	001156	5574#	6399																
SFILLS	001155	5574#	6399																
SGADR	001120	5574#																	
SGDAT	001124	5574#	5796*	5797	5799	5804*	5805	5807	5812*	5813	5815	5818*	5821*	5822*	5823*	5824*	5825*	5826*	

SGEY42	005532	6062#																	
SGTSHR	013336	6400#	6405																
SHD	000000	5550																	
SHIBTS	001000	5573#																	
SHIOCT	014142	6401#*																	
SICNT	001104	5574#	6391*																
SILLUP	012502	6394#																	
SINTAG	001135	5574#	6400																
SITEMB	001114	5574#	6392*	6393															
SLF	001172	5574#	6392	6399	6400														
SLFLG	014407	6402#*																	
SLPADR	001106	5574#	5696*	6391*															
SLPERR	001110	5574#	5696*	6391*	6392														
SHADR1	001226	5574#																	
SHADR2	001232	5574#																	
SHADR3	001236	5574#																	
SHADR4	001242	5574#																	
SHAITL	001174	5573	5574#	5696	6391	6392	6399												
SHAMS1	001224	5574#																	
SHAMS2	001230	5574#																	
SHAMS3	001234	5574#																	
SHAMS4	001240	5574#																	
SHBADR	001002	5573#																	
SHFLG	014406	6402#*																	
SHNEW	014031	6400#																	
SHSGAD	001210	5574#	6402*																
SHSGLG	001212	5574#	6402*																

SHSGTY	001174	5574#	6402*																	
SHSHR	014020	6400#																		
SHTYP1	001225	5574#																		
SHTYP2	001231	5574#																		
SHTYP3	001235	5574#																		
SHTYP4	001241	5574#																		
SHXCNT	012004	6391#																		
SHNULL	001154	5574#	6399																	
SHNST*	000001	5783#	5795#	5803#	5811#	5821#	5822#	5823#	5830#	5838#	5847#	5848#	5850#	5858#	5860#					
		5866#	5876#	5880#	5888#	5896#	5907#	5908#	5918#	5939#	5948#	5958#	5966#							
		5977#	6015#	6020#	6035#	6044#	6054#	6056#	6058#	6060#										
		6398#*																		
SHOCNT	013000	6398#*																		
SHOMODE	013002	6398#*																		
SHOVER	011770	6391#																		
SHPAS3	001202	5574#	5696*	6054	6056	6058	6060	6062*	6391											
SHPAS4	001006	5573#																		
SHWRAD	012476	6394#																		
SHWRON	012336	5696	6394#																	
SHWRMG	012472	6394#																		
SHWRUP	012410	6394#																		
SHQUES	001170	5574#	6392	6399	6400															
SHDOCHR	013550	6400#	6405																	
SHDOEC*	***** U	6405																		
SHDLIN	013670	6400#	6405																	
SHDOCT	014042	6401#	6405																	
SHDBZ	000010	6400#																		
SHDNAD	005554	6062#																		
SHRZA	***** U	6405																		
SHAVRE*	***** U	6405																		
SHAVR6	012506	6394#*																		
SHCOPE	011524	5696	6391#																	
SHETUP*	000117	5576#	5695#	5696	6062	6391	6392	6400												
SHSTUP	017777	5576#	5695#																	
SHVLAD	011734	6391#																		
SHVPC	000106	5571#																		
SHWR	0167400	5542#	5550	5556	5574	5696	5783	5795	5803	5811	5821	5822	5823	5830						
		5838	5847	5848	5850	5858	5866	5876	5880	5888	5896	5907	5908							
		5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054	6056	6058						
		6060	6062	6391	6392	6394														
		5574#	5696																	
		5556	6391																	
		6021#	6024	6032*	6306#															
		5574#	6391#																	
		5574#	5696*	5811*	5821*	5822*	5830*	5847*	5848*	5866*	5876*	5878*	5896*	5907*						
		5900*	5939*	5948*	5958*	5966*	5977*	6015*	6020*	6035*	6044*	6054*	6056*	6058*						
		6060#	6062#	6391*																
		5574#	6213	6400																
		5574#	6205	6400#																
		5543#	5550	5759	5783#	5795#	5800	5803#	5808	5811#	5821#	5822#	5823#	5828						
		5830#	5835	5838#	5847#	5848#	5850#	5858#	5866#	5876#	5880#	5888#	5896#	5908#						
		5885	5888#	5893	5896#	5907#	5908#	5910#	5936	5939#	5945	5948#	5954	5958#						
		5963	5966#	5972	5977#	5980	6015#	6017	6020#	6035#	6041	6044#	6050	6054#						
		6056#	6058#	6060#																

STPB	001152	5574#	6399																
STPPLG	001157	5574#	6399																

.HEADE	43#	5544#	5550	
.KT11	312#			
.8ETUP	1173#	5545#	5576	5695
.8WRM1	85#	5546#	5556	
.8WRLO	5556#			
.8ACT1	4953#	5548#	5571	
.8APT8	4997#	5548#	5574#	
.8APTH	5253#	5548#	5573	
.8APTY	5428#	5548#	6402	
.8ASTA	5299#			
.8CATC	898#	5544#	5557	
.8CMTA	1809#	5544#	5574	
.8DB2D	4583#			
.8DB2O	4706#			
.8DIV	4486#			
.8EOP	2155#	5544#	6062	
.8ERR0	2635#	5544#	6392	
.8ERRT	2830#	5546#	6393	
.8MULY	4423#			
.8PARM	5545#			
.8POME	4135#	5545#	6394	
.8RAND	4210#			
.8RDE	3806#			
.8RDOC	3715#	5547#	6401	
.8READ	3320#	5545#	6400	
.8RAZ	4850#			
.8SAVE	3881#	5545#		
.8SB2D	4667#			
.8SB2O	4768#			
.8SCOP	2389#	5545#	6391	
.8SIZE	4263#			
.8SPAC	5545#			
.8SUPR	4806#			
.8SWDQ	5545#			
.8STRAP	3983#	5545#	6405	
.8STYP8	3213#	5546#	6399	
.8STYPD	3136#	5546#	6390	
.8STYPE	2917#	5544#	5545#	6399
.8STYPO	3040#	5544#	6390	
.8QCA	937#			
.1170	491#			

ADC8	6389														
ADD	5718	5719	5720	5748	5982	5983	5984	5985	6186	6235	6263	6390	6393	6398	6399
ASL	6400	6401	6402	6401	6405										
ASLB	5986	6393	6400												
ASR	6390														
ASR	5818	5821	5845	5847	5873	5876	5903	5907	6031	6032	6187	6191	6192	6402	
BCC	6390														
BEQ	5696	5735	5776	5800	5808	5816	5821	5822	5828	5835	5843	5847	5848	5855	5863
	5871	5876	5878	5885	5893	5901	5907	5908	5918	5924	5930	5936	5945	5954	5963
	5972	5980	6028	6062	6156	6389	6391	6392	6393	6398	6399	6400	6401	6402	
BGE	6391														
BGT	6062	6390	6398	6400											
BMI	6391														
BIC	6026	6062	6165	6179	6214	6398	6400	6401							
BIS	5773	6161	6390	6392	6398	6400									
BISB	6169	6393													
BIT	6391	6392													
BITB	5696	6399	6402												
BLE	6232														
BLOS	6400														
BLT	6390	6398	6399	6400											
BMI	5751	5761	6206	6390											
BNE	5696	5709	5731	5733	5753	5757	5763	5819	5821	5822	5846	5847	5848	5874	5876
	5878	5904	5907	5908	6017	6033	6054	6056	6058	6060	6065	6147	6160	6164	6171
	6188	6205	6218	6216	6245	6390	6391	6392	6393	6394	6398	6399	6400	6402	
BPL	6105	6229	6298	6390	6392	6398	6399	6400							
BR	5687	5689	5696	5754	5759	5774	5789	6041	6058	6094	6098	6120	6124	6143	6166
	6212	6268	6279	6290	6389	6390	6391	6392	6393	6394	6398	6399	6400	6401	6402
CCC	6189														
CLC	6389														
CLR	5691	5692	5693	5696	5697	5705	5744	5745	5778	5779	5796	5824	5851	5881	5941
	5950	5960	5968	5987	6062	6181	6204	6262	6288	6390	6391	6393	6394	6398	6400
	6401														
CLRB	6390	6391	6399	6400	6402										
CMP	5696	5708	5752	5755	5790	5799	5807	5815	5821	5822	5827	5834	5842	5847	5848
	5854	5862	5870	5876	5878	5884	5892	5900	5907	5908	5917	5923	5929	5935	5944
	5953	6027	6215	6231	6390	6391	6400								
CMPB	5775	5979	6391	6392	6398	6400	6402								
DEC	6062	6159	6163	6170	6207	6209	6297	6393							
DEC8	6398	6399													
EMT	5551														
HALT	5557	6392	6394	6399											
INC	5686	5749	5978	5981	6062	6390	6391	6392	6394	6398	6400	6402			
INCB	6183	6391	6392	6399											
IDT	5551														
JMP	5557	5558	5559	5568	5563	5566	5698	5793	5908	6018	6062	6071	6088	6090	6093
JSR	5729	5780	6037	6048	6046	6049	6054	6056	6058	6060	6062	6083	6088	6089	6093
	6109	6114	6116	6119	6138	6142	6251	6253	6255	6287	6299	6272	6284	6287	6289
	6392	6399	6400	6402											
MOV	5688	5690	5696	5702	5703	5704	5706	5713	5714	5715	5716	5717	5743	5746	5766
	5772	5781	5784	5792	5794	5797	5798	5804	5805	5806	5811	5812	5813	5814	5821
	5822	5825	5826	5831	5832	5833	5838	5839	5840	5841	5847	5848	5852	5853	5859
	5860	5861	5866	5867	5868	5869	5876	5878	5882	5883	5889	5890	5891	5896	5897
	5898	5899	5907	5908	5911	5912	5913	5914	5915	5916	5921	5922	5929	5930	5933
	5934	5939	5940	5943	5944	5949	5952	5958	5959	5962	5966	5967	5969	5971	5977
	6013	6020	6021	6022	6024	6025	6035	6036	6039	6044	6045	6048	6054	6056	6058

MOV8	6060	6062	6067	6070	6075	6077	6078	6079	6080	6085	6089	6092	6100	6105
NEG	6106	6107	6108	6111	6115	6118	6130	6131	6132	6133	6135	6136	6141	6150
NOP	6162	6168	6177	6180	6182	6193	6203	6213	6224	6225	6226	6227	6233	6236
RESET	6239	6250	6252	6254	6256	6258	6271	6274	6275	6276	6277	6278	6283	6286
ROL	6293	6294	6295	6296	6309	6300	6391	6392	6393	6394	6398	6399	6400	6402
ROR	6405													
RTI	5696	6155	6157	6389	6390	6391	6392	6398	6399	6400	6401	6405		
RT8	5694	6390	6398											
SEC	5942	5942	5951	5961	5970	6062								
SUB	6389	6398	6401											
SWAB	6190													
TRAP	5696	6389	6390	6391	6392	6394	6398	6399	6400	6401	6405			
YST	5721	6099	6125	6148	6172	6194	6217	6236	6240	6266	6299	6393	6399	6402
	6389													6405
	5822	5848	5878	5908	6145	6146	6228	6390	6392	6402				
	6178													
	6405													
	5707	5730	5732	5734	5747	5748	5756	5760	5762	5785	5786	5787	5788	5822
	5878	5908	6016	6054	6056	6058	6060	6064	6264	6390	6391	6392	6393	6398
	6400	6401	6402	6405										5848
	6184	6205	6390	6391	6399	6400	6402							6399
	5574	6341	6342	6375	6378									
	5574	6062	6327	6328	6329	6330	6331	6332	6333	6334	6335	6336	6337	6338
	6340	6343	6344	6345	6347	6348	6349	6350	6351	6355	6356	6357	6358	6359
	6361	6362	6367	6368	6369	6370	6371	6393	6395	6400				6360
	6400													
	6398													
	5574	5768	5769	6062	6346	6374	6376	6377	6379	6389	6392	6398	6400	6402
	6400													
	4	5541	6400											
	6407													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821	5822	5823	5828	5830	5835	5838
	5847	5848	5850	5855	5858	5863	5866	5871	5876	5878	5880	5885	5888	5893
	5901	5907	5908	5910	5918	5924	5930	5936	5939	5945	5948	5954	5958	5963
	5972	5977	5980	6013	6017	6020	6028	6035	6041	6044	6050	6054	6056	6060
	6062	6094	6120	6143	6147	6389	6390	6391	6392	6393	6394	6398	6399	6401
	6402	6405												
	5551	6380	6393	6396	6402									
	5574													

.IIF	5550	5556	5557	5574	5696	6062	6070	6391	6392	6393	6399	6400	6405	
.IRP	5576	5695	5783	5795	5803	5811	5821	5822	5823	5830	5838	5847	5848	5850
	5866	5876	5878	5880	5888	5896	5907	5908	5910	5939	5948	5958	5966	5977
	6020	6035	6070	6054	6056	6058	6060	6390	6391	6392	6394	6401	6402	6013
	2	5537	5539	5540	5551	5556	5557	5574	5576	5695	5696	5727	5783	5795
	5811	5821	5822	5823	5830	5838	5847	5848	5850	5858	5866	5876	5878	5880
	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054
	6054	6060	6062	6247	6321	6391	6392	6400	6405					6056
	43	85	169	312	491	898	937	1009	1173	1239	1271	1295	1499	1471
	1519	1531	1578	1614	1647	1660	1681	1694	1718	1772	1839	1865	1922	2002
	2032	2088	2096	2155	2389	2635	2830	2917	3040	3136	3213	3320	3715	3806
	3983	4135	4210	4263	4385	4425	4486	4583	4667	4706	4768	4806	4850	4953
	5253	5299	5344	5428	5556	5574	5658	5672	5696	5990	6392	6364	6405	4997
	5544	5545	5546	5547	5548	5551	5574	5696						
	5574													
	5811	5821	5822	5823	5830	5838	5847	5848	5850	5858	5866	5876	5878	5880
	5896	5907	5908	5910	5939	5948	5958	5966	5977	6015	6020	6035	6044	6054
	6054	6060	6062	6242	6317	6391	6392	6400	6405					6056
	5574													
	5550	5551	5556	5557	5571	5573	5574	5576	5695	5696	5754	5759	5774	5776
	5709	5795	5800	5803	5808	5811	5816	5821</						